Topographic Visualization of the Internet

Pier Francesco Cortese, Giuseppe Di Battista, Antonello Moneta, Maurizio Patrignani and Maurizio Pizzonia

2006
Topographic Visualization of the Internet

P. F. Cortese, G. Di Battista, A. Moneta, M. Patrignani, and M. Pizzonia

RT-DIA-104-2005 Dicembre 2005

Dipartimento di Informatica e Automazione,
Università di Roma Tre,
Rome, Italy.
{cortese,gdb,patrigna,pizzonia}@dia.uniroma3.it
antonello.monetsen@gmail.com

Abstract

In the Internet thousands of Internet Service Providers (ISPs) exchange traffic according to their commercial agreements. ISPs are usually thought as organized in a hierarchy with the most important at the top. The peculiar business model of the Internet make paths that traverse higher levels of the hierarchy more expensive than paths that traverse only lower levels.

To visually represent routing paths traversed by traffic flows we propose a topographic metaphor which highlights the levels of the hierarchy they traverse. We propose an algorithm to compute such Internet topographic maps and show preliminary results. Our implementation is based on the BGPlay routing visualization system which is a web based monitoring tool widely used by ISPs.
1 Introduction

The structure of the Internet can be visualized at different abstraction levels for different purposes. For example, if we are interested in all the technical features we can look at the Internet at a very detailed level, showing the local area networks, the routers, and the point-to-point links. On the other hand, if we are interested in the relationships between Internet Service Providers (ISP) we can look at Internet at the Autonomous Systems (AS) level, where an AS is a collection of networks under the administrative authority of the same ISP. At such a level, Internet is currently partitioned into more than 20,000 ASes.

Many efforts have been done in the literature for visualizing the Internet at the AS level. A survey about the visualization of the Internet can be found in [?]. Some systems aim at visualizing the entire Internet structure (see e.g.[?, ?, ?, ?, ?]), while others visualize a little portion of it with different specific purposes. For example in [?] is described a system that visualizes the relationships between ASes that are known to an Internet Registry. Concerning “live” routing data, the BGPlay system [?] displays the evolution of the routing paths toward a specific AS over time. BGPlay is a quite popular tool for the ISPs and has been adopted by international organizations that publish real time collected data about inter-AS routing [?, ?]. A project that has a similar intent is shown in [?].

Recent studies [?, ?] have shown that the ISPs (and then the ASes) of the Internet are arranged into a customer-provider hierarchy (in the following Internet Hierarchy). In such hierarchy customers buy from their providers a transit service for their traffic to and from the Internet. In [?] each ISP is assigned to a level, according to its rank in the Internet Hierarchy. At this point, the problem of visually represent the Internet Hierarchy naturally arises.

An amazing number of methods have been devised to visually represent hierarchies in a variety of application domains (see, for example, [?, ?, ?]) many of which are targeted to show very large hierarchies.

The visualization of the structure of the Internet and of the Internet Hierarchy can be interpreted as a special case of a clustered graph drawing problem (first introduced in [?]), where the hierarchy describes a simple structure of clusters. Several authors dealt with the problem of representing a clustered graph using a spring embedder approach. Eades and Huang [?] proposed a system for the visualization of huge graphs, by first performing a clustering and then visualizing a portion of the graph by applying a force-directed approach. Walshaw [?] introduced an heuristic method for drawing large graphs which uses a multilevel technique combined with a force directed placement algorithm. Frishman and Tal [?] proposed an algorithm for dynamic drawings of clustered graphs.

In this paper we address the problem of the simultaneous visualization of the Internet Hierarchy and of other Internet features that are interesting from the application perspective. More specifically, we extend the visualization paradigm that have been successfully adopted in BGPlay [?] enriching it with support for the Internet Hierarchy visualization.

In Fig. 1 a screenshot of the BGPlay system is shown. The red vertex (AS137) is the AS we are focusing on. The picture represents paths to reach AS137 from several other ASes at a certain instant. For example, the path AS16150, AS6939, AS6762, AS137 is shown to be used for traffic incoming AS137 from AS16150. No information is available in this visualization about the relevance of ASes and their economic relationships.

Fig. 2 shows a screenshot of BGPlay enhanced with the visualization system presented in this paper. To visualize the Internet Hierarchy we use the metaphor of a topographic
map. The contour lines are used to confine ASes that are at the same level of the hierarchy. For example, the ASes inside the central brown area are top level ASes. As the “mountain” decreases in height the ASes decrease their rank in the hierarchy. The map shows quite well that some of the traffic flows have to climb the entire hierarchy to reach AS137, while other flows take “shortcuts”. For example, the path AS16150, AS6939, AS6762, AS137 do not pass through the Internet backbone since AS6939 and AS6762 exchange traffic at a lower level. Such kind of paths are usually more efficient and less expensive than paths that pass through the Internet backbone.

In this paper we present the following results.

- We propose a new metaphor for showing the Internet Hierarchy, the interconnection among the ASes and the routing at the AS level from specific observation points.
- We show a visualization algorithm that computes topographic maps of the portion of the Internet Hierarchy observed.
- We describe an implementation of the algorithm into the BGPlay system.

2 Background

In this section we provide networking background and a brief description of the BGPlay routing visualization system.

2.1 Networking

In the Internet each host is identified by an IP address (32 bits, usually written in the dotted notation, e.g. 193.204.161.48). An IP prefix identifies a set of (contiguous) IP addresses having the same leftmost \( n \) bits, with \( 0 \leq n \leq 32 \), and is usually written by attaching a /\( n \) at the end of the prefix (e.g. 193.204.0.0/15 indicates a prefix 15 bits long) [?]. Similarly to telephone call routing, inter-domain routing in the Internet is based on the destination prefix. Since a prefix identifies a set of addresses, it implicitly identifies a set of hosts having such addresses.

An **Autonomous System** (AS) is a portion of the Internet under a single administrative authority. In the Internet each AS is identified by an integer number. Traffic starting from an AS and directed to a specific prefix traverses an ordered set of ASes (**AS-path**). The configuration of such paths on the routing devices is too complicated to be manually performed. Hence, ASes exchange routing information with other ASes by means of a routing protocol called Border Gateway Protocol (**BGP**) [?, ?]. This protocol is based on a distributed architecture where border routers that belong to distinct ASes exchange the information they know about reachability of prefixes. Two border routers that directly exchange information are said to have a a peer session between them, and the ASes they belong to are said to be adjacent.

We define the **AS-graph** as the graph having one vertex for each AS and one edge between each pair of adjacent ASes. Note that, according to our definition, the ASes interconnection graph is not a multi-graph.

Each router stores information about routing in its routing information base (**RIB**). The RIB is a table where each row is a pair (prefix, **AS-path**) meaning that a certain
Figure 1: A screenshot of the BGPlay system.

Figure 2: A screenshot of the BGPlay system enhanced by the visualization approach described in this paper.
prefix is reachable through the associated AS-path. Such pairs are called routes. The main purpose of BGP is to allow the routers to exchange the routes they know. Since RIBs may be huge, the BGP process running on a router sends to its peers the full RIB only when a peering session is set up. During regular operation only changes to the RIB, termed updates, are sent.

A BGP update is either a route announcement or a route withdrawal. An announcement conveys the following information: “through me you can reach a certain prefix; to reach it, I will use the following AS-path”. A withdrawal nullifies a previously communicated route for a specified prefix. In other words a withdrawal means “you can no longer reach this prefix through me”.

A router which receives an update may or may not modify its routing table, depending on whether or not it knows routes which BGP considers "better" and depending on the routing policy of the AS itself. If the router modifies its routing table, it propagates the update to its peers.

Routes related to a certain prefix begin their existence within an AS called the originator of the prefix (typically the AS to which the prefix belongs). These routes are propagated by means of route announcements to adjacent ASes, which in turn propagate it to adjacent ASes. Every time a router propagates an announcement, it prepends its AS identifier to the AS-path; thus, the AS-path of an update is the list of ASes that the update has passed through.

2.2 Commercial Relationships Among ASes

ASes cooperate in order to ensure good connectivity service to their customers but are competitors from a commercial point of view [? , ?]. Commercial relationships among ISPs directly influence how BGP routers are configured. Router configurations affect, in turn, the AS-paths that can be announced in the Internet. A valid AS-path is usually made of three consecutive parts: (i) it first traverses the Internet walking on the edges of the AS-graph in the direction from customers to providers, (ii) it optionally traverses one peer-to-peer relationship between two ASes of the same relevance, and (iii) it traverses the Internet walking on the edges of the AS-graph in the direction from providers to customers till the destination is reached.

In an ideal world, tier 1 ASes do not have any provider and are involved in peer-to-peer relationships with all the other tier 1 ASes; further, tier-\(i\) ASes are customers of ASes of tiers greater than \(i\), are involved in peer-to-peer relationships with other tier-\(i\) ASes, and are providers of ASes of tiers less than \(i\). From BGP routing tables it is possible to infer the tier of the each AS [?].

2.3 The BGPlay Internet Visualization Service

BGPlay is a system that displays the portion of the AS graph that describes how the traffic flows to a certain AS from a set of selected ASes. It obtains routing data from well-known and publicly available sources, namely the Routing Information Service (RIPE NCC) [?] and the Route Views project (University of Oregon) [?] whose historical archives are used for network debugging purposes or scientific investigation and are updated in real-time.

BGPlay has a three tier architecture which permits its deployment over the web and easy access to several data sources. To query BGPlay, the user connects to a web page,
which hosts BGPlay, and starts the BGPlay applet which presents a query window asking for a prefix and a time interval. When the user submits the query, BGPlay processes the request and displays the animation window (Fig. 1), which presents the routing information.

The main part of the window shows an AS-graph and the routing at a certain instant of time for the prefix specified by the user. Each number represents an AS, and the AS originating the prefix is placed in the center of the graph and highlighted by a red circle. Each solid or dashed line represents a segment of an AS-path seen by RIS or Oregon Route Views. An AS-path starts in the originating AS and stops in the AS which provides BGP routing data to RIS or Oregon Route Views. The AS-paths that did not change during the query interval are merged into trees rooted at the origin AS and drawn dashed. Each tree is drawn in different color so that it can be unambiguously identified. AS-paths that did change during the query interval are drawn solid and are not merged.

BGPlay can animate the evolution of the routing over time. A panel, on the left of the window, shows the density of routing events over the interval of time of the query and highlights the instant of time whose routing is currently shown in the main part of the window.

3 Choosing the Visualization Metaphor

When we had to select a visualization metaphor for showing at the same time both the AS-paths of the BGP routing and the levels of the ASes hierarchy, we considered several possible options and focused on three main choices.

1. The classical approach of Sugiyama et al. visualizes a hierarchy by assigning a $y$-coordinate to each vertex (AS) according to its rank. In this way, top (bottom) level vertices appear in the upper (lower) part of the drawing.

2. Another strategy typically adopted for visualizing a hierarchy was to represent ASes with a size and/or color proportional to their rank in the hierarchy.

3. A further possibility was to extend the visualization approach already used for BGPlay by placing the ASes within a “topographic map” where each AS falls into the region assigned to its level.

We deliberately ruled out three-dimensional representations, since, apart from the fascination of such a scenario, effectively browsing a three-dimensional environment involves the use of sophisticated interfaces, while network operators and researchers appear to be reluctant to adopt particularly complex software architectures and tools.

Before discarding options 1 and 2 we performed several experiments against typical routing scenarios obtained by querying the Routing Information Service of the RIPE NCC and equipping this information with publicly available data on the Internet hierarchy. The purpose of the experiments was to compare the readability of the three graphic metaphors above from an information visualization perspective. To this aim we drew several diagrams according to the three strategies.

From our experiments we could observe the following pros and cons for the three metaphors.
Figure 3: A very preliminary handmade layered drawing showing a portion of the Internet Hierarchy along with a legend that associates colors to ASes. This approach was rejected since it leads to poorly readable drawing in most of the cases.
1. The traditional hierarchical representation through layered drawings, although very effective in conveying the hierarchy information associated with the ASes, proved to yield particularly entangled drawings. In fact, since usually AS-paths “climb” the hierarchy and then descend it, the drawings are very often cluttered by unavoidable crossings, which are known to greatly impact readability [?]. Figure 3 shows a small example of such drawings.

2. Using size and color to suggest the rank of the AS in the hierarchy has the advantage of being a very simple strategy, easy to realize and to plug into existing visualization systems, but, on the other hand has the major disadvantage of depriving the user of an overall view on the ASes of the same level. ASes are scattered in the drawing, and there is no clue of how many levels of the hierarchy are involved and how many links span more than one level.

3. The “topographic map” strategy produced the most clear and intuitive drawings. In such metaphor, ASes of the same rank are drawn inside the same region and regions corresponding to a higher level are contained inside the regions corresponding to the lower ones. Further, this metaphor is consistent with well-known drawing conventions in the area of inter-domain routing where the AS-paths are merged into a graph, which is colored in such a way to make it possible to uniquely reconstruct the path from each AS to the AS advertising the prefix.

More in detail, in our approach, the tiers of the ASes play the role of the elevation. We can imagine the drawing as a topographic map in which ASes of tier 1, the top level, are drawn in the middle, like the peak of a mountain, and ASes of lower levels are drawn progressively farther, as if they were the downhill of the mountain.

More formally, each tier $i$ is associated with a region of the drawing plane that we denote $A_i$ with $i = 1 \ldots T$, where $T$ is the number of tiers. Each region is connected, but regions $A_i$, with $i = 2, \ldots, T$, have one hole which contains all the regions $A_j$ with $j < i$. Region $A_1$ has no hole. All and only ASes of tier $i$ are placed into region $A_i$. The choice of the shape of regions $A_1, \ldots, A_T$ is critical since it constrains the layout of the ASes and hence may reduce the effectiveness of a layout algorithm in obtaining drawing with small edge length, even ASes distribution, and small number of crossings. To avoid the drawbacks of fixing the shapes of the regions we allow them to adapt to the ASes layout in order to improve readability. The overall effect, as produced by our algorithm described in Section 4, is a drawing in which the regions have highly irregular shapes which makes the drawing to look very much like a topography map with a superimposed AS-graph.

Concerning the representation of the AS-paths and of the way they change over time, we adopt the same drawing standard adopted by BGPlay and described in Section 2.3. An example of the final result is shown in Fig. 2.

4 Layout Algorithm

In this section we describe the layout algorithm which is based on a variation of the well-known force directed approach that is usually called spring embedder (see for example [?, ?]). In a conventional spring embedder the layout of a graph is obtained by simulating a physical system in which its vertices are electric charges that can move in the plane and
are subject to physical forces. Each charge is repulsed by all other charges. Each edge is a spring which has a length “at-rest” $l$. Each edge attracts its extremes if their distance is greater than $l$ and repulses them if their distance is less than $l$. The placement of the charges in the plane at equilibrium describes the final layout of the graph.

The input of our layout algorithm is the AS-graph where each AS is labeled with a tier number in $\{1, \ldots, T\}$.

The output of our layout algorithm is a layout of the AS-graph and the contours that separates areas $A_1, \ldots, A_T$ where each area contains all ASes associated with its tier as stated in Section 3.

Our layout algorithm is composed by two phases.

Phase 1. A set of $T$ concentric circles $C_1, \ldots, C_T$ are identified on the drawing plane, with radius $r_1 < r_2 < \ldots < r_T$. In this phase, these circles delimit the areas $A_1, \ldots, A_T$: $A_1$ is the area enclosed by $C_1$, $A_i$ (for all $i$ in $\{2, \ldots, T\}$) is the area enclosed by $C_i$ and outside $C_{i-1}$. At the end of this phase, for each tier $i$, all ASes belonging to tier $i$ are placed inside area $A_i$. In order to confine the ASes in the corresponding areas while keeping good readability of the layout, we adopt an incremental layout strategy that starts from ASes of tier 1 (see Figs. 5–7).

Consider, for each circle $C_i$ of radius $r_i$, circles $C'_i$ and $C''_i$ with radius $r'_i = r_i - \epsilon$ and $r''_i = r_i + \epsilon$ respectively, where $\epsilon$ is a very small number with respect to the radius of $C_i$.

1. ASes of tier 1 are confined inside $C'_1$ by a force that attracts these ASes toward the center of the drawing and that is null outside $C'_1$. ASes of tiers greater than 1 are confined outside $C''_1$ by a force that repulses these ASes from the center of the drawing and that is null inside $C''_1$. With these additional forces the spring embedder is run until equilibrium is reached.

2. The space between $C'_1$ and $C''_1$ is used to place a cycle of new vertices that is considered along with the AS-graph for the computation. We call this cycle a fence. Vertices of the fence cannot be moved in this phase but they interact with other vertices. This fence and the ones that will be added in the following will be the contours that separate adjacent areas in the final drawing. With the fence the spring embedder is run until equilibrium is reached. See Fig. 4.

3. Operations of steps 1 and 2 are executed also for tiers $2, \ldots, T$. That is, for each tier $i$, the spring embedder is run with new forces to contain inside $C'_i$ vertices of tiers less than or equal to $i$ and outside $C''_i$ vertices of tiers greater than $i$. A new fence is added and a new spring embedder run is performed. Throughout the computation, the forces introduced in the previous steps are not dropped, but remain to enforce proper confinement of each AS within its area. See Figs. 5, 6, and 7.

Phase 2. The layout is refined to obtain better readability: the contours separating consecutive areas $A_i$ and $A_{i-1}$ is left free to adapt homeomorphically while the layout of the AS-graph is improved for edge length and layout. This is implemented by allowing the vertices of all fences to move, and run the spring embedder algorithm until equilibrium is reached. Particular care is taken in order to avoid that fences intersect each other and that ASes broke the confinement of the fences. Since a fence in this phase varies its
perimeter, the density of its vertices along the contour may also dangerously vary leaving holes through which other vertices may pass. To avoid this problem, the density of the vertices of the fences is kept within a safe range: edges of the fences that are longer than a certain threshold are split and edges that are shorter than a certain threshold are contracted.

An example of the result of Phase 2 is shown in Fig. 8, while the corresponding picture presented to the user is shown in Fig. 2.

5 Implementation

The algorithm described in Section 4 has been prototypically implemented in the BGPlay system. No changes has been made to the BGPlay architecture described in [?]. Since the client tier is in charge of the visualization layout, this is the only part of the system affected by the integration of our topographic visualization approach. The information about levels of the ASes is at the moment taken from [?] and statically imported into the client.

The run of the algorithm takes approximatively 30-40 seconds to complete a layout depending on the AS-graph to show.

6 Conclusions

In this paper we dealt with the problem of visualizing the structure of the Internet at the AS level, together with information about the Internet Hierarchy. We introduced a new
Figure 5: Example of application of the algorithm. Phase 1, second fence

Figure 6: Example of application of the algorithm. Phase 1, third fence
Figure 7: Example of application of the algorithm. Phase 1, fourth fence

Figure 8: Example of application of the algorithm. Final layout after Phase 2. The actual picture shown to the user is presented in Fig. 2
visualization metaphor, and presented an algorithm to produce a layout for the AS-graph conforming to the described visualization metaphor. Our approach is to represent the AS-graph and the Internet Hierarchy as a topographic map, where the high level ASes, i.e. the Internet backbone, are placed at the top of the mountain.

This kind of metaphor allows an effective visualization of the routing paths that highlights the jumps they perform from a tier to another. The information about the tiers traversed by the routing paths might greatly help Internet Service Providers to adopt less expensive peering and traffic engineering strategies. Moreover, the topographic map represents a special case of clustered drawing and we believe that the techniques proposed in this paper can be successfully applied in different applicative contexts to represent a hierarchy of the vertices of a graph.

References


