

BGPlay: A System for Visualizing the Interdomain Routing Evolution*

(Long Demo)

Giuseppe Di Battista, Federico Mariani, Maurizio Patrignani, and
Maurizio Pizzonia

Università degli studi “Roma Tre”
Dipartimento di Informatica e Automazione
Via della Vasca Navale 79
00146 Roma, Italy

Abstract. In this paper we describe the visual interface of BGPlay, an on-line service for the visualization of the behavior and of the instabilities of the Internet routing at the autonomous system level.

A graph showing only connections among autonomous systems is not enough to convey all the information needed to fully understand the routing and its changes. BGPlay provides specifically tailored techniques and algorithms to show the routing at specific instants of time and to animate its changes. The system obtains routing data from well known on-line archives of routing information constantly kept up-to-date.

1 Introduction

The Internet is administratively partitioned into networks, called *Autonomous Systems (AS)*, where each AS is under a single administrative authority. Usually, each Internet Service Provider (ISP) controls one or more ASes.

Roughly speaking, at the AS level the Internet routing is described by a collection of sequences of ASes, called *AS-paths*. An AS-path $AS_1, AS_2 \dots, AS_n$ says that the packets directed to AS_n and originated by some device in AS_1 should traverse AS_2, AS_3 , etc. in the specified order. The AS-paths that give the full routing status of the Internet at a certain moment can be “merged” into a graph, called a *routing graph*.

To understand our goals, consider the Network Operating Center (NOC) of some ISP and suppose that the NOC wants to know what are the paths that the packets currently follow to reach AS_x which is operated by the NOC. BGPlay aims at showing the portion of the routing graph that describes how the traffic

* Work partially supported by European Commission - Fet Open project COSIN – COevolution and Self-organization In dynamical Networks – IST-2001-33555, by “Progetto ALINWEB: Algoritmica per Internet e per il Web”, MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by “The Multichannel Adaptive Information Systems (MAIS) Project”, MIUR Fondo per gli Investimenti della Ricerca di Base.

flows to ASx from a selected set of ASes which are considered, in some sense, “representative” of the entire Internet.

Further, suppose that the NOC is interested in understanding how the routing evolved during a specific time interval. This can be important for several reasons: to determine faults in the pipes surrounding ASx , to check the consistency of the routers configurations, or even to monitor the behavior of the partners of the considered ISP with the purpose to verify whether they fulfill the commercial agreements. BGPlay aims at showing all the routing changes “around” ASx that occurred during the prescribed time interval.

Instabilities and faults of interdomain routing have been the subject of recent research in the networking area (see for example [17,12,10,16]). Also, several works address the problem of effectively visualizing network related graphs [11, 13,20,5,1]

In this paper we describe the visual interface of BGPlay [7], which runs at <http://www.dia.uniroma3.it/~compunet/bgplay>. The purpose of BGPlay is the visualization of the behavior and of the instabilities of the Internet routing at the AS level.

BGPlay has a client-server architecture. The user performs a query on a Web browser (the client side) specifying an IP prefix and a time interval. The server identifies the AS that contains the given prefix and extracts information that describes the routing evolution in the given interval from the Routing Information Service (RIS) [2] of the RIPE and from a local mirror of the Oregon Route View (ORV) project [3]. RIS and ORV are large and well known repositories of routing information.

All the visualization problems are addressed on the client side by means of a java applet. The user interface exposed by the BGPlay client can visualize the routing graph detected at any instant of the specified time interval. The BGPlay visualization puts in evidence ASes interconnections and which AS-paths are active on that interconnections in a given instant of time.

The paper is organized as follows. In Section 2 we give basic interdomain routing concepts. In Section 3 we introduce the concept of *routing graph*, discuss the requirements for its visualization, and argue about the unfeasibility of some possible visualization approaches. In Section 4 we show the techniques adopted in BGPlay for the visualization of the routing graph. In Section 5 we show how BGPlay animates the evolution of the routing. Section 6 shows a typical session with BGPlay.

2 Networking Background

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 $0 \leq n \leq 32$. Such IP prefix is usually indicated attaching a $/n$ at the end of the prefix (e.g. 193.204.0.0/15 indicates a prefix 15 bits long) [22]. Routing in the Internet is prefix based (like for telephone call routing). Since a prefix identifies a set of

addresses, it implicitly identifies a set of hosts having such addresses. In the following the term *prefix* is used to denote both a set of addresses and a set of hosts, the distinction will be clear by the context.

An *Autonomous System (AS)* is a portion of the Internet under a unique administrative authority. In the Internet each AS is identified by an integer number. ASes cooperate in order to ensure good connectivity service to their customers but are competitors from an commercial point of view [14,15].

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*) [19,21]. Such a 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 perform a *peering session*, and the ASes they belong to are said to be *adjacent*.

Each router stores information about routing into its *routing information base (RIB)*. The RIB is a table where each line is a pair $\langle \text{prefix}, \text{AS-path} \rangle$ meaning that a certain 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 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 traversing a certain AS-path”. A withdrawal nullifies a previously communicated route related to a specified prefix. In other words a withdrawal means “you can no longer reach this prefix through me”.

The receiver of an update may or may not modify its routing table depending on whether the router knows routes which are considered better or not. If the router modifies the routing table, the update is propagated to its peers.

Routes related to a certain prefix “born” within an AS called the *originator* of the prefix. Then, routes are propagated to adjacent ASes, which prepend their AS identifiers to the AS-path of the route and propagate it again by means of route announcements.

3 Effective Routing Graph Visualization

The purpose of BGPlay is to provide a graphical representation of a portion of the Internet routing at a given instant of time and of the changes that affect routing over time. In this section we describe the requirements we considered and show some negative preliminary results.

We define a *collector-peer* as an AS that has a BGP peering with ORV or with RIS collectors. The collector-peers are the vantage points we use to inspect the routing in the Internet and are determined by the current configuration of

RIS and ORV. The set of collector-peers we are currently using is easy to identify by just looking at the ASes appearing as leftmost ASes in at least one AS-path among the AS-paths provided by ORV and RIS.

We focus the attention on a specific prefix that is in turn contained in one AS, the *target AS*. The *routing status* at a given time for that prefix gives for each collector-peer the AS-path representing the route chosen at that time by that collector-peer to reach the prefix. Such a status is effectively represented with a *routing graph*. A routing graph is a graph in which each vertex is an AS and edges are the pairs of ASes that appear consecutively in at least one of the AS-paths.

We have identified the following requirements for the visualization of the routing status:

- The attention of the user should mainly be focused on the target AS.
- An AS should appear in the drawing at a geometric distance from the target AS that is roughly proportional to the number of (AS-)hops separating them.
- For each visualized AS, the AS-path used to reach the target must be fully identifiable in the drawing, even if traversing edges that are traversed by other paths. Observe that this requirement would be easy to meet if the graph was a tree; in fact in this case there is just one tree path from each AS to the target AS. The presence of cycles makes the problem more complex.

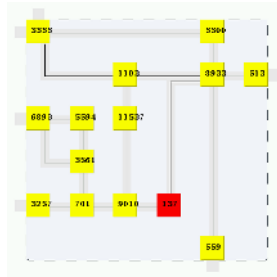


Fig. 1. In a first prototype of BGPlay the drawing standard used was orthogonal. The target AS (the dark node) was arbitrarily placed. The topological distance between AS137 and AS559 is two, while they are located very close in the drawing.

The choice of the drawing standard is biased by the first two requirements. Preliminary experiments we performed in this field have shown how the orthogonal drawing standard is not suitable for visualizing the routing graph. In fact, automatic layout tools in order to effectively compact the drawing may put the target AS in an arbitrary position and geometric distance among nodes does not convey any information about the topological distance (see Figure 1).

In the current version of BGPlay we adopt the straight-line drawing standard. For such standard there exist several layout algorithms that can fulfill the cited requirements (see Section 4).

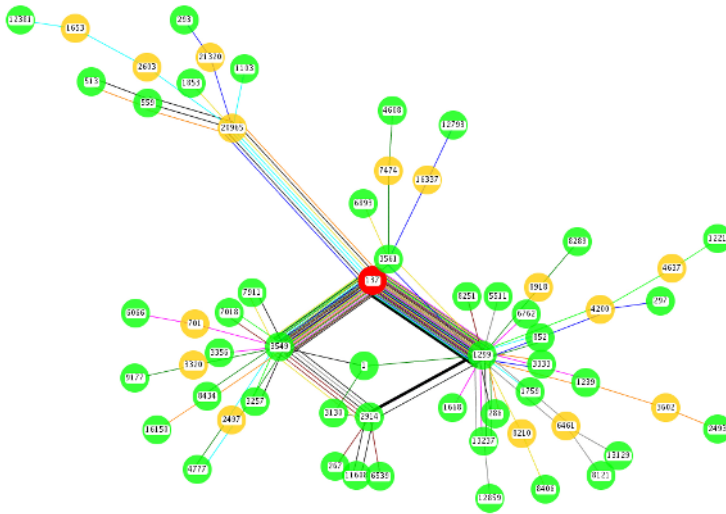


Fig. 2. In a first prototype of BGPlay all AS-paths were separately displayed. Paths were hardly distinguishable.

The third requirement is much more complex. Our first approach was to represent each AS-path separately and with a distinct color. A first prototype using this approach demonstrated that the number of paths to be shown was, usually, too high for the paths to be easily recognized due to the closeness of the paths and to the limited number of colors distinguishable on a monitor by human eyes (see Figure 2). The problem remains hard even using the techniques presented in [8] for drawing with fat edges.

Hence, we decided to adopt a different approach, described in Section 4.

4 Visualization Algorithms

In this section we describe the main algorithmic problems that we had to solve in order to design and implement BGPlay.

The third requirement stated in Section 3, that is that each AS-path must be easy to identify, is the one that was more difficult to meet. As we noted, if a set of AS-paths forms a tree, then such a set can be actually represented as a tree in an arbitrary color, satisfying the requirement. In fact, since each pair of vertices of a tree are joined by a single path, there would be no doubt about the sequence of ASes from the target AS to each collector-peer.

However, the routing graph is very often not a tree. Hence, our strategy is to partition the AS-paths into sets such that the graph obtained by merging the AS-paths of the same set is acyclic and can be unambiguously represented as a tree with a specific color. More formally, we have to solve the following problem.

Problem: TREE PARTITION

Instance: A set of AS-paths \mathcal{P} , such that each AS-path $p \in \mathcal{P}$ starts from a common AS_x .

Target: Find a partition for \mathcal{P} in k sets such that (i) the graph induced by the AS-paths in the same set is acyclic (ii) every partition for \mathcal{P} respecting (i) has at least k colors (that is, the number of sets is minimum).

Unfortunately, the TREE PARTITION problem is NP-hard. In order to prove this, following a standard technique, we show the NP-hardness of the corresponding decision problem, defined as follows.

Problem: K-TREE PARTITION

Instance: A positive integer K and a set of AS-paths \mathcal{P} , such that each AS-path $p \in \mathcal{P}$ starts from a common AS_x .

Question: Does a partition for \mathcal{P} in K sets exist such that the graph induced by the AS-paths in the same set is acyclic?

We prove that K-TREE PARTITION is NP-hard by reducing the GRAPH K-COLORABILITY to it. Recall that GRAPH K-COLORABILITY is an NP-complete problem defined as follows.

Problem: GRAPH K-COLORABILITY

Instance: A graph $G = (V, E)$ and a positive integer $K < |V| + 1$.

Question: Is G K -colorable, i.e., does a coloring of the vertices of G in K colors exist such that adjacent vertices have different colors?

Theorem. 1 *The problem K-TREE PARTITION is NP-hard.*

Proof. We reduce GRAPH K-COLORABILITY to K-TREE PARTITION. Given an instance of GRAPH K-COLORABILITY we construct the corresponding instance of K-TREE PARTITION as follows. For each vertex v of $G(V, E)$ we introduce an AS-path p_v . At the beginning of the construction all AS-paths have length one, and contain the same AS_x . Then, for each edge $(u, v) \in E$, we append to the two AS-path p_u and p_v the same $AS_{(u,v)}$. It is easy to show that the construction of the K-TREE PARTITION instance can be made in polynomial time, and that a solution for the instance of the GRAPH K-COLORABILITY problem exists iff a solution for the original instance of the K-TREE PARTITION does. \square

Because of Theorem 1, in BGPlay, to solve TREE PARTITION, we used the following greedy algorithm which runs in polynomial time.

The sets of AS-paths are denoted S_0, S_1, \dots, S_{m-1} , where the current number of sets is m .

1. A compatibility matrix is computed in which each AS-path is compared with each other. Two AS-paths are incompatible if and only if they form a cycle.
2. Only one empty set exists at the beginning: $m = 1$ and $S_0 = \phi$
3. foreach AS-path p

- 3.1 foreach set i from 0 to $m - 1$
 - if p is compatible with all paths in S_i then accommodate p into S_i , skip the rest of this cycle and continue with the next path, otherwise try the next set S_{i+1} if it exists.
- 3.2 if p has not been accommodated in any of the available sets add a new set S_m , initialize $S_m = \{p\}$ and increment m .

For the correctness of the algorithm it is crucial to observe that all the paths have a common endpoint (target AS). The worst case time complexity of the algorithm described above is $O(n^2)$ where n is the number of paths (considered bounded in length). In fact, the number of compatibility tests performed for each AS-path is at most n .

An edge traversed by more than one tree is displayed using as many lines as the number of trees traversing that edge, where each line is colored with the color of the corresponding tree. Fig. 3 shows a drawing produced by BGPlay. It is about a prefix announced by AS702. Three levels of gray are used. AS-paths 16150 8434 3549 702 and 1853 1239 702 are drawn in gray. The other paths are grouped in two trees drawn in black and and light-gray. Only dashed lines are used since, as we shall see in Section 6, BGPlay uses solid lines to highlight paths involved in routing changes.

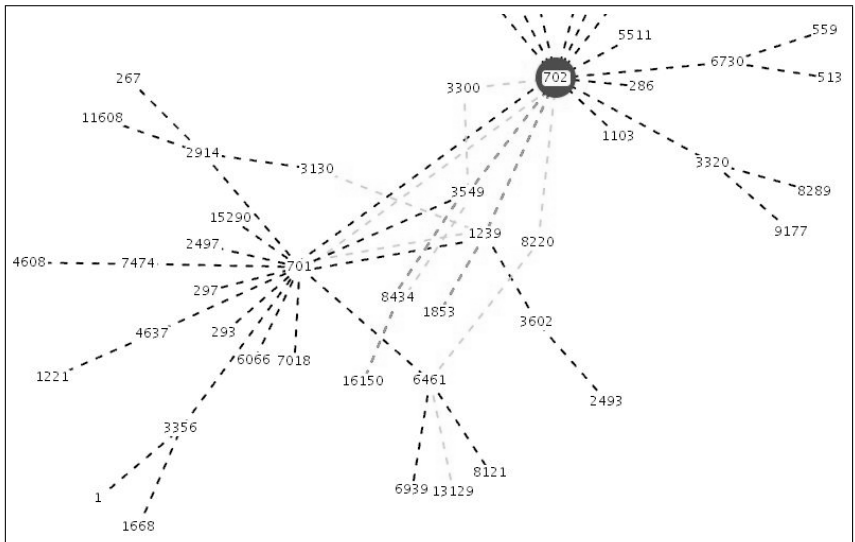


Fig. 3. A drawing produced by BGPlay. Three levels of gray are used. AS-paths 16150 8434 3549 702 and 1853 1239 702 are drawn in mid-gray. The other paths are grouped in two trees drawn respectively in black and light-gray.

The drawing layout is computed by using a spring embedder [6]. The target AS is constrained to be in the center of the drawing area. The computation is

performed for a fixed number of steps that are much larger of the steps needed in the average case by a drawing to reach equilibrium.

What we said up to now concerns the visualization of the routing graph at a given instant of time. However, even for a single user query, BGPlay needs to visualize several routing graphs, each corresponding to the status of the routing at a specific instant.

Obviously, the visualization of the routing status before and after a BGP update occurred is sufficient to convey the information of the update. However, it is essential for two consecutive visualizations to be similar for not changing too much the users “mental map” [9,18].

We address the above problem by computing, before any visualization or animation takes place, the layout on a graph that is the union of all the AS-paths that appear in any routing graph at any instant of the selected interval. When the animation takes place, at any instant we display only the edges that belong to the current routing graph.

This trick ensures that consecutive animation steps show drawings that are very similar, since node positions are the same. Further, nodes involved in route changes (see Section 5) are most of the times placed near. In fact, two AS-paths involved in a routes change begin and end with the same AS, hence, the spring embedder tends to draw them closely.

5 Animation of the Routing Changes

Obviously, the visualization of the routing status before and after a BGP update occurred is sufficient to convey the information of the update. However, it is essential for two consecutive visualizations to be “similar” [9,18]. Further, the routing change should be apparent to the user.

A *routing history* is given by a starting routing status and a sequence of routing changes. From a routing history it is possible to reconstruct the routing status at each instant covered by the routing history.

A routing change happening at instant t can be one of the following types.

New route. A collector-peer that is not connected at time right before t with the target AS now acquires connectivity using a specified AS-path.

Route change. A collector-peer which is connected at time right before t with the target AS using AS-path p changes its connectivity by using AS-path q with $q \neq p$.

Route withdrawal. A collector-peer which is connected at time right before t with the target AS, loses its connectivity.

Further, it is interesting to consider the following event that is not a routing change but can be a symptom of network problems.

Route re-announcement. A collector-peer is connected at time t with the target AS and an announcement is received for the same route as if the route was new.

The collector-peers are partitioned in stable and unstable peers according to the following rules:

- if during the period of time covered by the considered routing history a collector-peer is continuously connected with the target AS using the same AS-path, that collector-peer is considered stable,
- in all other cases the collector-peer is considered unstable.

To each stable collector-peers (and its stable AS-paths) a color is assigned according to the greedy algorithm described in Section 3. Distinct colors are assigned to all unstable collector-peers.

The union of all AS-paths that appear in the considered routing history forms a graph G in which the nodes are the involved ASes. A layout of the nodes of G is performed by using a spring-embedder algorithm as in Section 3. Such a layout remains unchanged during all the animation.

Animation is performed in the following way.

- AS-paths related to stable collector-peers are drawn dashed with their colors as in Section 3 and are not affected by any animation.
- AS-paths for new routes appear in their regular position (according to the layout of G). The attention of the user is caught by making the thickness of new AS-paths pulsing.
- Route changes are animated by means of a polyline morphing from the old AS-path to the new AS-path.
- Route withdrawals catch the attention of the user by a thickness pulse of the involved AS-path before it disappears.
- Route re-announcements catch the attention of the user by a thickness pulse of the involved AS-path.

The polyline morphing used to show route changes could be realized in several ways (see for example [4]). In BGPlay we adopted a rather simple technique which is illustrated in Figure 4. Let p be the polyline representing the old AS-path and p' be the polyline representing the new AS-path. A bijection between points of p and points of p' is defined in the following way. Polylines p and p' are consistently oriented incoming the target AS. Points in p are mapped with real numbers x in $[0 \dots 1]$ preserving the ordering. The same operation is performed on p' . Morphing between points which have the same value of x is performed linearly both in space and in time. In our case the two extremes of a polyline are always the same, namely the target AS and the collector-peer. The figure shows also the shape of the polyline in three intermediate instants. Morphing time is approximately set to one second.

6 Monitoring the Routes of the Traffic Incoming a Given AS with BGPlay

In this section we show how a user (e.g. the NOC of an ISP) can exploit the BGPlay service to monitor the evolution of the routing around an AS manager by the ISP.

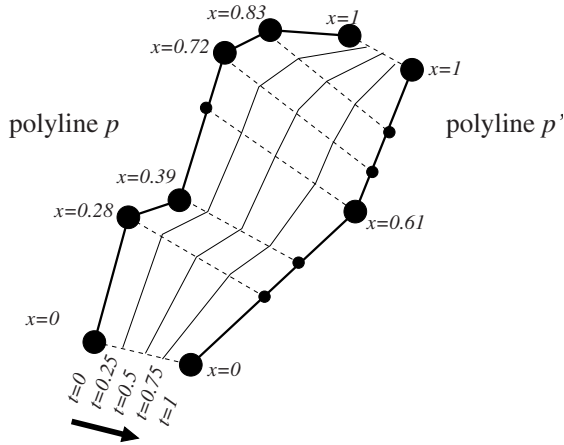


Fig. 4. Route changes are animated using a linear polyline morphing.

Welcome to Flap Viewer v.2.0 (beta)
 This tool is designed to show the instabilities in BGP routing of a specific prefix

Insert here the prefix that you want to explore:

Prefix: . . . /

Time Interval:
 Select the time interval of your query:
 - to have Route Views data your starting time must be less than a week ago
 - to have Ris data the query starting time must be less than three months ago

Starting ti...
 Date / / Time : :

Ending time:
 Date / / Time : :

Data Sources:
 Choose the Data Sources that you want to "use" in your query. RIS data are taken from on line user interface, while Route Views data come from local cache

<input checked="" type="checkbox"/> RIS RRC00	<input checked="" type="checkbox"/> RIS RRC05	<input type="button" value="Clear"/> <input type="button" value="Canc..."/> <input type="button" value="OK"/>
<input checked="" type="checkbox"/> RIS RRC01	<input checked="" type="checkbox"/> RIS RRC06	
<input checked="" type="checkbox"/> RIS RRC02	<input checked="" type="checkbox"/> RIS RRC07	
<input checked="" type="checkbox"/> RIS RRC03	<input checked="" type="checkbox"/> RIS RRC08	
<input checked="" type="checkbox"/> RIS RRC04	<input checked="" type="checkbox"/> Route Views	

Fig. 5. The starting form.

Suppose that the NOC is interested in monitoring the routing evolution concerning the prefix 193.0.0.0/21 from May 21, 2003 to May 23, 2003, because in that period some network instabilities have been somehow perceived. She/He fills the form of Fig. 5. Observe that the user is filling out the form selecting all the available sources of information (checkbox RRC00, RRC01, etc.).

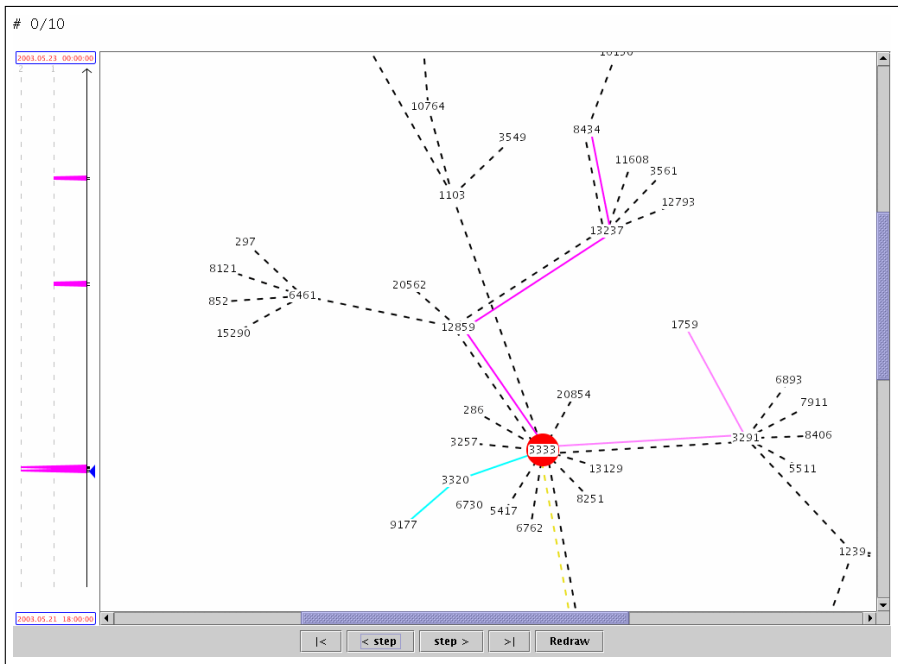


Fig. 6. A routing graph shown by BGPlay.

BGPlay shows (Fig. 6) a routing graph that represents the routing status at the time of the first event in the selected time interval. Observe that the AS originating the prefix is 3333 and it is placed in the center of the window. The paths that represents stable routes are drawn dashed while the paths associated to unstable routes are drawn solid.

The buttons on the bottom allow the user to move through the sequence of events that happened in the specified time interval. Both forward and backward moves are possible. A time panel (on the left) shows the distribution of the routing events in the interval and an arrow indicates the time of the event just displayed.

Now, the user steps through the events. BGPlay shows the the evolution of the routing. The status bar (on top) indicates the event identifier, a timestamp, and the type of the event (see Section 5). Depending on the type, additional information is displayed:

New route. The AS-path of the new route.

Route change. Both the old and the new AS-path.

Route withdrawal. The AS-path that is no longer valid.

Route re-announcement. The AS-path that is re-announced.

References

1. Hermes. <http://www.dia.uniroma3.it/~hermes/>.
2. Routing Information Service of the RIPE (RIS). <http://www.ripe.net/ripencc/pub-services/np/ris/>.
3. University of Oregon RouteViews project. <http://www.routeviews.org>.
4. S. Bespamyatnikh. An optimal morphing between polylines. *International Journal of Computational Geometry & Applications*, 12(3):217–228, 2002.
5. A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the high level structure of the internet with hermes. *J. of Graph Algorithms and Applications*, 6(3):281–311, 2002.
6. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.
7. G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia. Archives of bgp updates: Integration and visualization. In *Proceedings of IPS 2003, International Workshop on Inter-domain Performance and Simulation*, pages 123–129, 2003. online.
8. C. A. Duncan, A. Efrat, S. G. Kobourov, and C. Wenk. Drawing with fat edges. *Lecture Notes in Computer Science*, 2265:162–177, 2002.
9. P. Eades, R. F. Cohen, and M. L. Huang. Online animated graph drawing for web navigation. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 330–335. Springer-Verlag, 1997.
10. L. Gao and J. Rexford. Stable internet routing without global coordination. In *Measurement and Modeling of Computer Systems*, pages 307–317, 2000.
11. R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *IEEE INFOCOM 2000*, pages 1371–1380, Tel Aviv, Israel, March 2000.
12. T. Griffin and G. T. Wilfong. An analysis of BGP convergence properties. In *SIGCOMM*, pages 277–288, 1999.
13. B. Huffaker, D. Plummer, D. Moore, and k claffy. Topology discovery by active probing. Technical report, Cooperative Association for Internet Data Analysis - CAIDA, San Diego Supercomputer Center, University of California, San Diego, 2002.
14. G. Huston. Interconnection, peering and settlements – part 1. *Internet Protocol Journal*, 2(1):2–16, 1999.
15. G. Huston. Interconnection, peering and settlements – part 2. *Internet Protocol Journal*, 2(2):2–23, 1999.
16. C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. In *SIGCOMM*, pages 175–187, 2000.
17. C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, 1998.
18. K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Lang. Comput.*, 6(2):183–210, 1995.
19. Y. Rekhter. A border gateway protocol 4 (BGP-4). IETF, RFC 1771.
20. N. Spring, R. Mahajan, and D. Wetherall. Measuring isp topologies with rocketfuel. In *Proceedings of ACM/SIGCOMM '02*, Aug. 2002.
21. J. W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, Reading, MA, 1999.
22. A. S. Tanenbaum. *Computer Networks*. Prentice-hall International, Inc., 1996. ISBN: 0-13-394248-1.