Visualizing Internet Routing Changes

Mohit Lad, Student Member, IEEE, Dan Massey, Senior Member, IEEE and Lixia Zhang, Fellow, IEEE

Abstract—Today’s Internet provides a global data delivery service to millions of end users, and routing protocols play a critical role in this service. It is important to be able to identify and diagnose any problems occurring in Internet routing. However the Internet’s sheer size makes this task difficult. One cannot easily extract out the most important or relevant routing information from the large amounts of data collected from multiple routers. To tackle this problem, we have developed Link-Rank, a tool to visualize Internet routing changes at the global scale. Link-Rank weighs links in a topological graph by the number of routes carried over each link, and visually captures changes in link weights in the form of a topological graph with adjustable size. Using Link-Rank, network operators can easily observe important routing changes from massive amounts of routing data, discover otherwise unnoticed routing problems, understand the impact of topological events, and infer root causes of observed routing changes.

Index Terms—network visualization, information visualization, Internet routing, interactive graphics, data analysis, visual mining.

I. INTRODUCTION

Today’s Internet provides a global data delivery service to millions of end users. Network routing protocols play a critical role in this delivery service by steering data traffic towards their destinations. Effective diagnosis tools are imperative to enable network operators to identify routing problems in this global system. Several diagnosis tools, such as traceroute and BGPlay[1], are available for analyzing routing changes regarding a single destination. However, a fiber cut may change the routes to a large number of destinations, resulting in significant network traffic movement which may in turn trigger routing dynamics in other areas. Thus, it is essential to be able to observe network routing changes at the Internet scale to understand the overall impact of a single topological event.

To this end, we have developed Link-Rank, a tool to visualize routing changes in the global Internet. Not only can a picture capture the meaning of “thousands of words” but it can also lead to instant comprehension. However a fundamental challenge facing the Link-Rank design is how to capture routing changes in a comprehensive visual picture, given the sheer size, the topological complexity, and the highly dynamic nature of the Internet routing system. Millions of routing updates are generated daily and there is no easy way to extract information about most important or most relevant routing changes. All the existing single destination diagnosis tools utilize a specific starting point and a given destination to trace the routing path or the path changes. To examine routing changes at large, however, one does not have a clear starting or ending point to focus on. Instead, one is facing a topology with over 20,000 networks (Internet Autonomous Systems) and 180,000 destination entries. We need a new conceptual model that can capture the network behavior of aggregate routing changes.

Link-Rank extracts the total number of routes carried over individual links in the Internet topology, called link weight, and measures the changes in the number of routes on each link, as a way to capture aggregate routing changes. To reduce the data size to a comprehensible level, Link-Rank uses an input-filter to extract the most important or relevant routing changes from the large amount of routing data. To enable network operators to quickly spot potentially problematic time periods for further investigation, Link-Rank provides an activity plot that summarizes routing changes along the time dimension. Link-Rank also offers the user an output filter to adjust the display density in visualizing routing dynamics. Using case studies, we show how the above features provided by Link-Rank can help network operators mine and understand interesting routing changes from gigabytes of routing data.

The remainder of this paper is organized as follows. We first review the relevant background of Internet routing in Section II. We then introduce the design of Link-Rank in Section III, where we discuss the design challenges, describe our solutions, and explain in detail several useful features of Link-Rank. In Section IV, we show the utility of Link-Rank to network operators by using Link-Rank to discover and understand large scale routing changes. In Section V, we discuss the impact of Link-Rank on network research and operations. In Section VI, we review related work in the area of network visualization, and in particular, compare Link-Rank to two other tools BGPlay and ELISHA. Finally, in Section VII, we present possible directions to proceed for future work.
II. BACKGROUND OF INTERNET ROUTING AND BORDER GATEWAY PROTOCOL

The Internet consists of a large number of networks called autonomous systems (AS). Each AS is assigned an AS number and contains one or multiple destination networks. Each destination network is represented by an IP address prefix. For example, the prefix 131.179.96.0/24 represents a network at UCLA and is part of AS 52 (UCLA’s AS number). As of March 2006, the Internet consists of over 20,000 autonomous systems and over 180,000 prefixes.

A routing protocol propagates the information about how to reach all the destinations, throughout the network. A path vector protocol called Border Gateway Protocol (BGP) [2] is the de-facto routing protocol used between autonomous systems in the Internet today. Routing information in BGP is propagated by the exchange of BGP update messages. A BGP update message contains information about the destination prefix and the AS path used to reach that prefix. We represent a BGP update in the form \( \{ \text{prefix} : \text{AS-path} \} \). Figure 1 shows how BGP updates propagate routing information in the Internet. In this figure, AS 22 owns a prefix P1 and sends a BGP update message \( \{ P1 : 22 \} \) to its neighbor AS 33. AS 22 is said to be the origin AS for prefix P1. On receiving this update, AS 33 now prepends its own AS number to the received path and sends the BGP update \( \{ P1 : 33, 22 \} \) to its neighbors, AS 44 and AS 55.

AS 55 in turn sends the BGP update \( \{ P1 : 55, 33, 22 \} \) to its neighbor AS 44. Note, AS 44 receives two paths to reach P1. When an AS receives more than one path to reach a prefix, it chooses one of them as the primary path. In Figure 1, we assume AS 44 picks the path \( \{ P1 : 33, 22 \} \) because it is shorter. Generally speaking, this decision on which path to pick is based on the routing policy of each individual AS. An AS’s routing policy also determines whether to send a particular path to a neighbor. Besides initial route propagation, physical events like link failures can also trigger BGP updates. For example, assume the link (44, 33) goes down. As a result, AS 44 switches to a backup path \( \{ 55, 33, 22 \} \) that it knew earlier and sends the BGP update \( \{ P1 : 55, 33, 22 \} \) to its neighbors.

To control the number of BGP updates and thus reduce processing on routers, it is recommended that BGP routers set a BGP timer called MinRouteAdver timer to a value of 30 seconds. This timer sets the minimum time a router needs to wait before sending BGP updates to its neighbor for the same destination. In other words, in the case above, when AS 33 sends the BGP update \( \{ P1 : 33, 22 \} \) to its neighbors, it would have to wait at least 30 seconds before sending another update for the same prefix P1.

Since BGP updates propagate routing information in the Internet, capturing the BGP updates at various parts of the Internet can give us useful insight of the state of the Internet and amount of routing changes going on in the Internet. In Figure 1, AS 44 is connected to a routing update collection box that receives BGP updates from AS 44. This collection box represents the data collectors of BGP monitoring projects such as RouteViews [3] and RIPE [4]. We call an AS connecting to such a collection box as an observation point. These monitoring projects collect BGP updates from various observation points (operational routers in autonomous systems) around the globe and make the data available to the public. This data can then be used by network operators and researchers for various tasks such as routing problem identification and diagnostics. However, due to the large size of the Internet topology, millions of BGP updates are generated everyday, contributing to the large volume of updates collected by RouteViews and RIPE. In the remainder of this paper, we show how we can visualize the routing changes conveyed by these millions of BGP updates.

III. VISUALIZATION DESIGN

The fundamental objective of Link-Rank is to visualize routing changes. A major challenge we faced in this regard is scale, i.e. over 180,000 destinations and 20,000 AS nodes. In addition, one has to deal with the large number of BGP updates. For example on April 1, 2006 we observed over 250,000 updates from a single observation point, AS 7018. To deal with this issue of scale, in Link-Rank we take the approach of weighing links by routes carried, regardless of where the destinations of these routes are. By assigning these weights, we are able to visually represent heavily used links in the form of Link-Rank graphs as well as capture...
changes in these weights in the form of Rank-change graphs described in Section III-A.

Link-Rank uses an input filter to control generation of Rank-change graphs. Input filters described in Section III-B can be threshold based like ‘construct Rank-change when weight of a link changes by more than 50’, or ‘show changes of routes only to specific set of prefixes’. To provide a summary of amount of routing changes, in Section III-C, we introduce activity plots that summarize routing activity over time. Activity plots are very useful starting points to identify time periods of high routing dynamics. Finally, one may want to control the level of time granularity to observe route changes that last longer than a certain amount of time. Such granularity control can be achieved by using time windows and drill down features explained in Section III-D.

A. Rank-change Graph

The Link-Rank graph from an observation point weighs a link by the number of routes using that link. This notion of ranking a link with number of routes translates to the name Link-Rank. In the Internet, a single AS cannot see the complete Internet topology nor can it know the routes taken by all the other ASes. Thus, the weight associated on the edge in a Link-Rank graph is relative to the observation point and does not tell us how many total routes in the entire Internet use this link. To explain this concept, we use a simple example shown in Figure 2. This figure depicts the routing table seen by a router in AS 44 in Figure 1a, in the form of a graph. Here we assume the existence of two more prefixes P2 and P3 announced by AS 33 and AS 55 respectively. In Figure 2a, link (44, 33) has a weight of 2, since that link appears twice in the routing table at AS 44. We denote the link weight by $wt((\text{link}), (\text{observation-point}))$, e.g. $wt((44, 33), 44) = 2$. We define Link-Rank graph from a node as a graph showing all the links along with weights used by that node, like Figure 2a. Note that the direction of the link is important in a Link-Rank graph. If BGP updates received at AS 44 change the routing table at AS 44, the Link-Rank graph will also change. In Figure 2b, AS 55 withdraws its route to P3, and as a result of this withdraw message, AS 44 shifts to an alternate path to reach P3. The weight of link (44, 33) has now increased from 2 to 3. In reality, a Link-Rank graph from a BGP router can have close to 20,000 links, and hence entire Link-Rank graphs are difficult to visualize.

To understand BGP dynamics, we need to understand how many links change weights as a result of the BGP updates. As a first step, we looked at BGP updates over a period of one week and marked the links changing rank after each BGP update. We found that the changes usually came in bursts. As a result, instead of looking at the Link-Rank graph after each BGP update, we could analyze just two Link-Rank snapshots, the one before the burst of updates and the one after the burst of updates. We also found the burst of updates to affect the weights of a much smaller set of links in most cases. Rank-change graphs capture these links whose weights have changed.

A Rank-change graph takes the difference between two Link-Rank graphs and uses red (or dashed) edges to mark the links that have lost routes and green (or solid) edges to mark links that have gained routes. Simply stated, given two Link-Rank graphs from $G_1$ and $G_2$ at different times $t_1$ and $t_2$ respectively, a Rank-change graph plots all links $(a, b)$ where the weight on these links $wt((a, b), G_1) − wt((a, b), G_2) ≠ 0$. Figure 3a shows the Rank-change graph for the routing change in Figure 2. From this figure, one can clearly see that link (44, 55) lost 1 route, while the link (44, 33) and (33, 55) gained a route. Note, the Rank-change graph does not show links that have not gained or lost routes, e.g. link (33, 22). A Rank-change graph can either show only link weights, only weight changes or both. For example, Figure 3a shows just the weight changes, while Figure 3b shows the current link weight followed by the weight change in parenthesis.

1) Nodes, edges and color coding: We now discuss some details of visualization in Rank-change graphs. Figure 4 shows an actual Rank-change graph from BGP data. Note,
the Internet has over 20,000 autonomous systems, and currently only a few hundred observation points are connected to public data collectors. Observation points from where one can observe routing changes, are shown as circular nodes to differentiate them from rectangular nodes that are not observation points. Visually separating the observation points from the other nodes, clearly highlights other possible view points that can be used to better understand the same time interval. The observation point of the Rank-change graph (AS 6453) is colored blue to differentiate from other observation points that are colored orange.

Edges in Link-Rank are primarily red or green in color. An edge is colored red when it loses routes and green when it gains routes. To help users with difficulty to distinguish between certain colors, Rank-change graphs can also be displayed using dashed and solid lines to indicate loss and gain, instead of red and green. In addition, this representation is very useful in the process of assembling multiple views explained in Section III-F. The thickness of the edges in the Rank-change graph represents the magnitude of weight change. With links of varying thickness, one can easily spot links with high losses or gains. In addition to varying the edge thickness, the size of the nodes varies based on the amount of weight change of edges and the number of such edges adjacent to it. This scaling of nodes helps to identify ASes with high routing activity.

We use the JUNG visualization library [5] to construct the Rank-change graph. Link-Rank uses the spring layout implementation from the JUNG library, which gives satisfactory results in general. Furthermore, the layout implementation also allows one to manually reposition any node as needed for clearer view. In most cases when the Rank-change graphs were sparse, the users of Link-Rank were satisfied with the default layout. With denser graphs, the users tended to reposition some nodes. Some user reactions to the input and ideas for improving the layout are discussed in the Section VII on Future work.

B. Components of Link-Rank

The three components of the Link-Rank tool are shown in Figure 5. An important component is the input filter block that controls when the Rank-change graphs are constructed. In Figure 3, we saw the Rank-change graph for a single route change. In reality, input filters are needed to enable Link-Rank to scale in regard to topology size and number of BGP updates. One input filter involves picking a specific set of prefixes and examining the routing changes for these prefixes. Another input filter is a threshold based scheme and is the filter used in all our case studies explained later in this paper. In this threshold based scheme, we maintain the instantaneous link weight for each link in the topology seen by an observation point. In addition, we maintain the change in weight since the last Rank-change graph was generated. The link weight as well as the change in weight is updated for all links affected by each BGP update message. A Rank-change graph is generated when the weight of any link changes by more than a preset threshold (default is 50). A detailed treatment of this scheme and numerical results of the effect of threshold is beyond the scope of this paper and the interested reader may find more details in [6].

Using the threshold filter with BGP updates, a single routing event may be broken into multiple Rank-change graphs. For example, assume a link \((A, B)\) fails and 5000 routes using that link are affected. This will result in a burst of 5000 BGP updates closely spaced in time, each of which reduces the rank of the link \((A, B)\) by 1. Thus the entire update burst would reduce the rank of \((A, B)\) by 5000. If the threshold filter generates a Rank-change graph each time the link weight changes by 50, there would be as many as 100 Rank-change graphs, each with a change of 50 routes on link \((A, B)\). We employed a timing mechanism to reduce the number of Rank-change graphs due to the same event. We observed that by delaying the construction of the Rank-change graph by a short time, we could drastically reduce the number of Rank-change graphs due to the same event. We call this time to delay construction of Rank-change graph, as event timer and set its value to 30 seconds. During the event timer, if routing changes add weight \(x\) to a link and immediately change back to reduce the weight on that link by \(x\), the net weight change would be 0 (termed as compensating change) and hence no Rank-change graph will be generated (since weight change is below threshold of 50). Our choice of 30 seconds for the event timer was motivated by the BGP timer called MinRouteAdver timer explained in Section II. With the MinRouteAdver timer
set to the recommended time of 30 seconds, compensating changes cannot happen at a frequency less than 30 seconds. Though not all routers in the Internet are known to use the MRAI timer, we found the event timer value of 30 seconds to be adequate.

The graph generator component outputs the Rank-change graph based on the updates fed to it by the input filter. The output filter can control the links and nodes in the Rank-change graph for brevity. Filter rules for the output could be simple weight based rules such as ‘remove all links below a change of 100’ or more complex such as ‘show graphs with at least one of the nodes 338, 55 AND links 44→33’. The output filter is part of the visualization tool, and based on graph complexity, one can dynamically use filter rules to simplify the graphs. Summarizing, the input filter prepares the data for Rank-change graphs and the output filter can be used to prune the Rank-change graph further.

C. Activity plots: summarizing weight changes

Activity plots summarize routing changes represented by Rank-change graphs along the time dimension. An activity plot is a series of red and green bars on alternate sides of a horizontal axis of time. With an activity plot, a user can identify time periods of high routing activity and then investigate those specific periods in more detail. We first explain how a single activity bar is plotted. Figure 6 shows a Rank-change graph similar to Figure 2. Given a Rank-change graph, we first find the total gain and total loss by adding the weight changes of the green and red links respectively. In this case, the total rank gain is 200 (100 each on links (44, 33) and (33, 55)) and the total rank loss is 100. We plot red and green bars proportional to the total loss and gain respectively as shown in Figure 6. In this case, the green bar is longer than the red bar. A higher gain (green) than loss (red) could be due to a combination of longer new paths as in Figure 6 and new routes being announced.

Activity bars can provide summary information about the routing change. For example, if we only see a red bar, it signifies that routes have been lost entirely and this means some set of prefixes are not reachable.\(^3\) In an activity plot, one activity bar is constructed for each Rank-change graph over the duration of the activity plot. The total magnitude of the activity bar could vary a lot depending on the type of event, and we adjust the scale for the Y-axis, where the highest magnitude in any interval coincides with the tallest bar on the activity plot and the remaining bars scaled linearly relative to this. In Section IV, using case studies, we illustrate how activity plots can help in the identification of routing problems.

D. Time Windows and Drilling Down

The time window control in Link-Rank allows users to aggregate Rank-change graphs in a time interval. Due to the presence of slow convergence [7], some short lived invalid paths could appear as genuine route changes. With the time-window control, one can increase or decrease the longevity of weight changes that one wants to visualize. Figure 7a shows three activity bars corresponding to three Rank-change graphs shown below. In Figure 7b we show the time window by rectangular boxes on the activity plot. This time window can slide along the activity graph using DVD playback like controls. In Figure 7b, we show how the Rank-change graph looks in three cases, two involving the same time window size but different positions, and one involving an even wider time window size. At each position of the time window, the Rank-change graphs falling in that window are combined into one by taking the union of all the Rank-change graphs. Equivalently, the Rank-change graph for a specific position of the time window can also be constructed as a difference graph between the Link-Rank graphs at the start and end of the time window. Note that within the first time window t1, the Rank-change graphs have some cancellation effect of route changes, i.e. net weight change of (44, 55) is \(-100 + 50 = -50\). In contrast, within the second time window t2, the Rank-change graphs have an additive effect, i.e. weight change of (44, 55) is \(+100 + 100 = 200\). If the time window is increased to include all three activity bars as in t3, then all the changes will be cancelled and the net Rank-change graph will be empty.

Another time control, called the drill-down feature allows one to control the time granularity of the entire activity

\(^3\)There are cases where a red bar and absence of green bar may not reflect prefix loss. E.g. if the paths for a set of prefixes change from \(A \rightarrow B \rightarrow C\) to \(A \rightarrow B\) because the prefixes are now originated by B, link \((B, C)\) loses ranks but prefixes may still be reachable.
Fig. 7. Use of time window to control time of change

plot. By drilling down, one can expand the activity inside the current time-window to a larger time-span in a new window. The first part of Figure 8 shows an activity plot spanning over 8 days and time window of 16 hours. To better understand the activity inside the time window, we drill down to expand the 16 hour time window to the activity time span in middle activity plot in Figure 8. The time window in this case is about 2 hours. Drilling down further on this time window will expand these two hours further as shown in the last activity plot. One can now see the individual activity bars in detail compared to the first activity plot. Note, given an activity plot, one can drill down to the granularity of the time equal to the event timer explained in Section III-B.

E. Pruning Rank-change graphs

Link-Rank processes BGP updates and visualizes the links that have changed. In all the examples in this paper, the underlying network consists of the Internet with about 20,000 nodes. However, the size of the Rank-change graph depends on the number of links that have changed and the magnitude of changes. Hence in some cases where a small number of links have changed, the Rank-change graphs may contain only a small number of nodes and links. In other cases with a lot of changes, the Rank-change graph may contain hundreds of nodes, making it difficult to extract information visually. Link-Rank allows a user to prune Rank-change graphs using different filtering techniques to reduce the complexity of the graph. One technique to prune the graph by using an output filter in the form of a threshold filter to remove edges with weight change value less than a threshold value set by the user. Other types of filters include viewing the top N links with highest weight change values, and view links adjacent on a set of user specified AS. One can also use a combination of all these filters and specify the order in which filters are applied.

F. Assembled View: Merging Rank-change graphs from multiple observation points

Link-Rank views from multiple observation points can be assembled in a single Rank-change graph. Figure 9 shows the assembled view from two observation points AS 11608 and AS 3561. Note, here we have to use the dashed and solid lines to indicate lost and gained routes. Edges in this example are either blue or pink, blue indicating the changes from AS 11608, while pink indicating the changes from AS 3561. In general, in assembled views, each observation point and its changes are represented by a unique color. With assembled views, one can identify common segments of change in Rank-change graphs across different observation points and narrow down on the possible cause of
In this section, we use examples to show how Link-Rank can be used to discover and analyze routing events.

A. Methodology

Our objective is to evaluate how Link-Rank can help network operators discover and diagnose routing problems. In terms of routing data, network operators have access to BGP routing tables and update messages received at their routers. We have access to similar data from the public archives of the RouteViews Oregon collector that contains routing tables and updates from about 40 routers belonging to different autonomous systems. In order to understand how network operators diagnose problems, we interacted with network administrators through email and personal interviews at various North American Network Operator Group’s meetings [8]. Our pool of interviewees consisted of about 40 operators from both small and big ISPs, most of them having more than 5 years of experience in network operations. In the rest of this section, we use the knowledge gained from this interaction to analyze three case studies from the perspective of an operator using Link-Rank.

We used three ways to select observation points and time periods for case studies. First, we looked at activity plots from all observation points on a weekly basis and identified the periods with dense activity or spikes. Case I is an example of this, where we saw some heavy activity from a particular observation point. Second, we looked at activity plots to find common activity spikes across multiple observation points during the same time period. Case II is an example where activity plots from multiple observation points show spikes at around the same time. Cases I and II show that activity plots can serve as summaries for network operators using Link-Rank. Finally, we picked case studies in response to reports of routing or traffic problems from external sources such as North American Network Operators Group (NANOG) mailing lists. Case III is representative of this category where there were reports of traffic problems from a few ISPs. In each of these cases, we used the Rank-change graphs during the selected time periods, and in one case assembled multiple views together, to understand the routing activity.

B. Case I: Capturing Link Instabilities

Around March 2005, AS 7018 showed a lot of heavy activity as shown in the second activity plot (router IP 12.0.1.63) in Figure 10 showing activity for a period of one week. One task of the network operator is to find out whether this activity is because of a problem within AS 7018 or a problem beyond AS 7018. Another question to be answered is whether the entire activity is due to the same event or different events. We drilled down the activity from one week to a one hour period on March 9, 2005 shown in Figure 11. Note from Figure 11 showing activity over one hour, that a Rank-change graph was generated almost every minute.

We then looked at the Rank-change graphs in this period and found a common sequence of changes. Figure 12 shows a typical sequence of Rank-change graphs we found, with the time window set to 1 minute. This Figure shows that 134 routes switched between the paths 7018 \(\rightarrow\) 80 and 7018 \(\rightarrow\) 1239 \(\rightarrow\) 80. This behavior was observed for
almost three weeks in March 2005. Next step was to find out the preferred path among the two oscillating paths. From examination of routing tables before the event, we saw that the preferred path to reach AS 80 was the direct link (7018, 80). Since the weight of the link (7018, 80) on the preferred path repeatedly touched 0, it seemed likely that the link between AS 7018 and AS 80 went up and down repeatedly and was the cause of the instability seen.

Events such as the constant route change above may result in longer delays as well as possible packet losses. Yet, they often go unnoticed. In this case, the behavior continued for almost 3 weeks in March 2005 contributing hundreds of thousands of BGP updates seen at the observation point. A network operator using Link-Rank at AS 7018 would benefit from the quick identification of such oscillations and bring stability to routes as well as reduce the number of BGP updates in the Internet drastically. In our examination over other time periods, we found quite a few instances of link instabilities similar to this case above.

Summary: Densely clustered bars in activity plots, especially where they have near constant height are almost always a strong indication of link instabilities. Activity plots are useful in spotting such cases. One can then examine these time periods in detail to figure out the actual causes of the rapid route changes.

C. Case II: Root-cause identification

Root cause identification involves inferring the cause of an observed set of routing updates. For Case II, we picked a case where activity plots of many observation points showed spikes around the same time. Figure 13 shows the activity plot of a few observation points from October 18, 2005 to October 24, 2005. One can easily spot spikes and dense activity in these plots from multiple observation points (around October 21, 2005). To understand the causes, we looked at the routing activity from AS 6453 (router IP 195.219.96.239) which generated the first activity plot in Figure 13. Starting from an entire day’s activity, we drilled down to a four hour period between 4:00 and 9:00 GMT on October 21, 2005 that contains the dense activity. Figure 14 shows this Rank-change graph around 06:20 GMT on October 21, 2005 from AS 6453 with a time window set to 15 minutes. During this time, link (6453, 3356) lost close to 3000 routes (out of a total of around 140,000). At the same time, some other links like (6453, 701) and (6453, 1239) gained routes. Note, for ease of presentation, we do not show the link weights and prune the graph by applying the filter to remove links with changes less than 200. Based on observation, the possible cause is either AS 6453, AS 3356, or the link (6453, 3356).

In this case, since similar activity is also seen from other observation points, one can benefit by combining multiple observation points into a single assembled view. Figure 15 on page 9 shows the assembled view from three observation points, AS 6453, AS 1239, and AS 3257 that showed similarity in activity plots. In the assembled view, we use dashed lines to represent route loss and solid lines to represent route gain and assign each observation point and its corresponding changes, a unique color, e.g. AS 3257 and its corresponding changes are colored blue. The orange colored nodes indicate other potential observation points, so more views can be added. Here we select only three observation points to make the Rank-change graph easy to understand. After we reduce the time window to 5 minutes, one can see from Figure 15, multiple links to and out of AS 3356 were affected, strongly suggesting some problems inside the AS 3356 and not just the link between AS 6453 and AS 3356. Our observation was validated by reports from the NANOG discussion forum that AS 3356 indeed had some internal problems, and was further corroborated by discussions with network operators.

Summary: To use Link-Rank for identifying root cause, one can look for high loss or gain links or nodes which have a high number of outgoing edges with weight changes.
One can also assemble multiple views along the lost or gained path to isolate sections of the path which might be problematic.

D. Case III: Detecting and visualizing prefix hijacking

Our final case study was picked in response to reports of routing problems on mailing lists and network operator forums. On December 24, 2004, customers of AS 6939 reported that they were unable to reach many Internet sites. However, the routing table from AS 6939 did not show any noticeable reduction in number of entries, implying that routes were still reachable. If routes to all sites still existed, what else would have caused inability to reach the sites? By looking at activity plots, we saw a spike around the time of complaints as shown in Figure 16, an indication routing activity going on.

We plotted the activity for Dec 24, 2004 and drilled down to the time between 8:30 and 10:30 UTC. Figure 17 on page 11 shows the Rank-change graph from AS 6939 around 9:15 UTC with a time window of 15 minutes. Notice...
the difference in the characteristic of this graph. In typical cases of route changes, there is one source node where the edges with weight changes, start, and one or more sink nodes where the weight changes converge. For example in Figure 12, the source node is AS 7018, while the sink is AS 80, while in Figure 14, the source node is AS 6453, while some sinks are AS 22773, AS 18566 and AS 6389. Note that in both case I and case II, most of the sink nodes have a red as well as a green incoming edge. This convergence of a red and green edge on a common node is because the origin AS for a prefix does not change in general. If the source and origin remain the same, the red (old) and green (new) paths converge on some common node between the old and new path. In this case however, AS 6939 saw routes added on a single path 6939 → 6762 → 9121, but the sinks did not show a convergence of both a red and green edge. This implied that the old and new paths did not share a common origin AS as is usually the case. On examining the routing table, we saw thousands of prefixes having routes with AS 9121 as the origin AS. Before this event, these prefixes had various different origin ASes. So clearly, while the routes still existed to reach the destination prefixes, the routes were invalid and hence the traffic got black-holed at AS 9121. An event of this type where an AS wrongly advertises prefixes it does not own, is referred to as a prefix hijack and is considered as a serious security threat to the Internet.

Following messages from the NANOG discussion forums, and after consulting with various network operators, we confirmed that AS 9121 originated almost all the prefixes in the Internet, thus making a route through AS 9121 more lucrative than some of the longer but genuine routes. We saw similar impacts on other observation points with the effect of this hijack varying based on routing policies of observation points and how far they were from AS 9121. While it may seem that such events can be automatically detected, the key purpose served by visualization here is to highlight the source and extent of this hijack attack to the operator. In Section VII, we discuss some directions for providing hints for known event characteristics like the prefix hijack case mentioned here.

Summary: A visual characteristic of large scale prefix hijack events is the lack of red (lost ranks) and green (gained ranks) edges converging on the sink nodes. Any Link-Rank graph showing such characteristics should be a cause for alarm.

In this section, we presented three case studies. In each of these cases, we show how Link-Rank can be used to discover the problem as well as identify the cause of the problem.

V. DISCUSSION

The Internet routing infrastructure is a big and complex system. The large volume of BGP log data makes it difficult for network operators to observe and understand BGP dynamics. As network researchers, we faced the same challenge and we have developed Link-Rank as a visualization tool to aid our work in network routing research. Link-Rank developed the concepts of link weight and Rank-change graphs as a simple yet effective ways to capture routing changes. The input filtering mechanism prepares data for visualization and the output filtering mechanism controls what to display. The time window controls the longevity of weight changes in a Rank-change graph. Finally, the activity graphs provide a summary of routing changes for quick scan.

We have used Link-Rank to identify various problems. For example, when a BGP session is established and broken down (termed as BGP session reset) repeatedly, routes oscillate as shown in case I in Section IV. Using Link-Rank we identified potential BGP session resets by looking for links whose rank drops to 0. We further identified several cases of BGP sessions that were unstable for long periods of time resulting in hundreds of thousands of updates. This ability of Link-Rank to present visuals summarizing routing changes spread across thousands of routing updates and allowing the operators to use their expertise to interpret the visuals makes it a very useful operational tool.

We feel Link-Rank will definitely have an impact on research in BGP. For example, identifying the underlying event triggering the routing updates, called root cause analysis, has been an active area of research in the Internet routing community for the past few years. Link-Rank captures changes in routes carried by links and we feel the use of Link-Rank for root cause identification has great promise. The values of link weights and weight changes as well as characteristics of the Rank-change graph like the number of red and green edges flowing into and out of a node can help identify potential root causes. We believe Rank-change graphs will help in developing new methodologies to address the problem of root cause identification.

VI. RELATED WORK

In this section we discuss related work in the area of visualization applied to the networking domain.

A. Visualization of Route Dynamics

In the area of visual analysis of Internet routing, BGPlay [1] shows changes in routes from different monitors to a particular prefix. BGPlay visualizes an update stream and uses animation to highlight the change in routes. A tool closely related to BGPlay is ELISHA [9] [10]. This work has a similar flavor to BGPlay and analyzes events on a
per prefix basis. In this scheme, updates on a prefix are sequentially arranged next to a time line and a line is drawn from the time line to the updates. This helps in easily identifying effect of the updates clustered close together. One can then delve into the details of a particular event, by visualizing the path changes in the form of an arc based representation of links in the routing paths with each AS being assigned a unique X coordinate. This visualization can help in understanding the updates as well as detecting routing anomalies. Both BGPlay and ELISHA complement Link-Rank. Note that BGPlay and ELISHA capture events to a particular destination, while Link-Rank visualizes aggregate routing changes affecting multiple prefixes. Thus, on detecting routing problems to a specific prefix using BGPlay or ELISHA, one can use Link-Rank to see if the problem is related to some link level issues and vice versa.

Other closely related work to Link-Rank is detecting prefix hijacks using visualization [11]. The main difference lies in the fact that [11] provides a visual technique for detecting abnormality in prefix announcements but does not tell which ASes get affected as shown in detail by Link-Rank. Interesting events exposed by visualization in [11] could be investigated using Link-Rank to understand the event impact.

B. Visualizing connectivity and anomalous behavior

Another area of application of visualization to networking is visualizing network connectivity. [12] visualizes the router level connectivity, while [13] provides a tool for interactive visualization of AS level connectivity.

Networking research and operations has also benefited from visualization of traffic flows to detect intrusions and anomalies. [14] enables one to visually identify anomalous and potentially harmful events like port scans and failed login attempts. Visualizing port level activity alone can lead to good anomaly detection and is the focus of PortVis: [15]. NVisionIP [16] is another tool used to visualize port scans and host scans. NIVA [17] is a haptic based system that can be fed data from a commercial intrusion detector in order to make it more usable for network operators. VisFlowConnect [18] is a tool designed to identify anomalous traffic patterns and can visually capture events like virus outbreaks and denial of service. Other security based visualization work includes [19] and [20].

C. Knowledge discovery in Internet routing

Finally, a lot of research has also been done on examining BGP logs to discover problems and patterns. Most of the work here has been on inferring the root cause of observed BGP updates. Various heuristics have been presented and applied to public data from RouteViews [21], [22], [23]. These works present interesting approaches to root cause inference, but lack the ability to involve an expert directly in the inference. An interesting line of work would be to incorporate these heuristics to the visualization tools like BGPlay, ELISHA and Link-Rank.

Other related work has been in the area of understanding instabilities in BGP [24] and behavior of BGP under stress events such as worms [25]. These works attempt to understand and classify BGP updates received from observation points. For example, updates are classified into ones where a path is withdrawn, and a new path is announced. Visual plots like number of updates in hourly or daily bins, and
counts for each of the update classes help in understanding which class of updates saw a rise under a known event. This work provides a good understanding of the BGP updates, but does not describe the effect or cause of these BGP updates. Link-Rank on the other hand provides visual information allowing one to summarize events in terms of which routes changed.

VII. Future Work

On the visualization front, we are exploring ways of improving the node layout in the Rank-change graph. Some users expressed the desire to assign position constraints to selected nodes in the Rank-change graphs. We also observed that users often repositioned nodes to separate the green paths from the red paths. Incorporating position constraints and color of edges as input to the layout algorithm are interesting directions for future work. Another direction to deal with denser graphs is to be able to bring a sub graph to the forefront. In particular we are exploring the idea of selecting an AS and bringing to the forefront its connected components. Yet another line of work involves better distinction of contributions from each observation point. Currently, Link-Rank relies on colors especially when assembling views from multiple observation points into a single graph. Users with difficulty in differentiating colors would benefit from other ways to represent different views in the same graph. In activity plots, due to the Y-scale adjusted based on changes within the time of the activity plot, it is difficult to easily compare activity plots from different observation points with each other. We are exploring ways to enable easier comparison between activity plots from multiple observation points.

Besides visualization, we feel Link-Rank can also benefit from built-in event recognizer and classifiers. We showed how Rank-change graphs can be used to identify different kinds of events like link problems, AS events and prefix hijack. We are working on using simple rules to generate signatures of events and match Rank-change graphs to these known signatures.

Acknowledgments

We would like to acknowledge Yiguo Wu and Timothy Ma for implementing the visualization aspects of Link-Rank. Jason Chen and Stuart Wood also contributed to the very first version of Link-Rank tool. We would like to thank many members of the North American Network Operators Group (NANOG), UCLA’s Internet Research Lab and Colorado State University’s NetSec Lab for their feedback during the development of Link-Rank. We would also like to thank the anonymous reviewers of IEEE TVCG for their insightful comments and suggestions that have helped improve this paper.

References

[19] Robert Ball, Glenn A. Fink, and Chris North, “Home-centric visualization of network traffic for security administration,” in


Mohit Lad received his Bachelor’s degree in Computer Engineering from Mumbai University, India in 2000, and Masters degree in Computer Science from University of California, Los Angeles in 2003. Currently, he is pursuing a Ph.D. degree in the Computer Science at the University of California, Los Angeles. His current research interests include fault diagnosis and monitoring in large scale networks, Internet routing security and information visualization. He is a student member of IEEE and ACM.

Dan Massey is an assistant professor at Computer Science Department of Colorado State University and is currently the principal investigator on DARPA and NSF funded research projects investigating techniques for improving the Internet’s DNS and BGP infrastructure. He received his doctorate from UCLA and is a senior member of the IEEE, IEEE Communications Society, and IEEE Computer Society. His research interests include fault tolerance and security for large scale network infrastructures.

Lixia Zhang received her Ph.D in computer science from the Massachusetts Institute of Technology. She was a member of the research staff at the Xerox Palo Alto Research Center before joining the faculty of UCLA’s Computer Science Department in 1995. In the past she has served as the vice chair of ACM SIGCOMM, Co-Chair of IEEE Communication Society Internet Technical Committee, and on the editorial board for the IEEE/ACM Transactions on Networking. Zhang is currently serving on the Internet Architecture Board.