Mapping and Visualizing the Internet

Bill Cheswick
Bell Laboratories
ches@bell-labs.com

Hal Burch
Carnegie Mellon University
hburch@cs.cmu.edu

Steve Branigan
Bell Laboratories
sbranigan@bell-labs.com

Abstract

We are collecting and recording routing paths from a test host to each of over 90,000 registered networks on the Internet. We've collected this data systematically since August 1998. The resulting database contains interesting routing and reachability information, and is available to the public for research purposes. The daily scans cover approximately a tenth of the networks on the Internet, with a full scan run roughly once a month. We have also been collecting Lucent's intranet data, and, for a time, we collected paths to Yugoslavia during and after the war.

To visualize these databases, we use a simulated spring-force algorithm to lay out the paths. This algorithm is well known, but has never been applied to such a large problem. The Internet graph, with around 88,000 nodes and 100,000 edges, is much larger than those previously considered tractable by the data visualization community. The resulting Internet layouts are pleasant, though rather cluttered. On smaller networks, like Lucent's intranet, the layouts present the data in a useful way. For the Internet data, we have tried plotting a minimum distance spanning tree; by throwing away data, we can make the remaining data more accessible.

Once a layout is chosen, it can be colored in various ways to show network-relevant data, such as IP address, domain information, location, ISPs, etc.

1 Introduction

Network administrators have long used Van Jacobson's[?] *traceroute* to identify the path taken by outgoing packets towards a given destination. Each "hop" on the outgoing path is a router, and most routers will respond to a *traceroute*-style packet with the IP address of one of its network interfaces. An average Internet path is 17 hops, and our maximum shortest path is 25.

By obtaining a list of all announced networks on an internet, and discovering the path to each of these networks, we can build a good picture of the "center" of the Internet, and a kind of picture of what the Internet looks like as a whole. Of course, this is an egocentric view, as it only captures the paths taken by our outgoing packets. Thus, we obtain a reachability graph, not a complete map.

We quickly discovered, however, that Mapping is a more generally useful pursuit, as it became obvious that mapping an intranet is valuable. Large intranets are hard to manage, and offer many security problems. A map can yield a lot of information and can help spot likely leaks in a company's perimeter security. This work led to the intranet perimeter work described in [?].

Each morning we scan Lucent's intranet and the Internet itself. On Lucent's intranet, we run full scans daily down to the class C network level. On the Internet, our daily scans cover about one tenth of our list of possible destinations, reaching each announced network several times a month. We run full scans of the Internet about once a month, and trim the database of older path data every few months to keep its size manageable. The daily Internet scan data and labels available on a web page[?] and saved to CD-ROM. We plan to run these scans for years.

We also ran scans to the class C network level of Yugoslavia during the bombing in the spring of 1999, to see if we could spot the effects of the bombing on that country's internet. The data obtained from our mapping resulted in several interesting features that could be related to bombing activity.

Due to the magnitude of the resulting databases, visualization is a must. The eye can help us gain some understanding of the data we are collecting. We can pick out interesting features for further investigation and find errors in Internet router configurations, such as routers that return invalid IP addresses. We'd like to have a large paper map with the properties of traditional flat maps: they can help one navigate towards destinations, determine connectivity, readily reveal major features and interesting relationships, and are hard to fold up.[1]

We use a spring-force algorithm to position the nodes on the map. A few simple rules govern the adjustment of a point's position based on proximity of graph neighbors, number of incident edges, and the number and position of close nodes which are not neighbors. We shuffle these points for 20 hours on a 400MHz Pentium to obtain the maps shown in this paper. Sample graph sizes are:

	networks	edges	nodes
Lucent	3,366	1,963	1,660
Internet	94,046	99,664	88,107

The next section describes the motivation behind the collection of the data. Section ?? gives detail about how the mapping is done, and section ?? discusses the layout heuristics. The results of the Yugoslavia mapping during NATO bombing are presented in section ??.

2 Motivation

The initial motivation for collecting path data came out of a Highlands Forum, a meeting that discussed possible responses to future infrastructure attacks using a scenario from the Rand Corporation. It was clear that a knowledge of the Internet's topology might be useful to law enforcement when the nation's infrastructure is under attack. Internet topology could also be useful for tracking anonymous packets back to their source.

An openly available map could be useful to monitor the connectivity of the Internet, and would be helpful to a variety of investigators. In particular, it might be useful to know how connectivity changes before and during an attack on the Internet infrastructure.

Good ISPs already watch this kind of information in near real-time to monitor the health of their own networks, but they rarely know anything (or care much) about what that status of networks more than one hops away from theirs is. There is no one responsible for watch the whole Internet. Of course, no one can watch the entire Internet because it is much too large. There is a major web of interconnecting ISPs that in some sense defines the "middle" of the net—the most important part.

Our current attempts, using *traceroute* packets, only map outgoing paths, and only from our test host—we discuss these limitations later. Even this limited connectivity information can yield insights about who is connected to whom.

The database itself can be useful for routing studies and graph theorists looking for real data to work with. Since we are collecting the data daily over a long period of time, we may be able to extract interesting trends. We can leverage our system administration skills to try to collect systematic data daily, building a consistent database.

The mapping software has lent itself to another pressing problem: controlling an intranet. Software that can handle 100,000 nodes on the Internet can easily handle intranets of similar size. An intranet map can be colored to show insecure areas, business units, connections to remote offices, etc. The work in [?] was originally suggested from the mapping data.

Our visualizations of the Internet itself have attracted wide media interest. Most people visualize the Internet by showing people staring at a web browser. Our maps give some idea of the size and complexity of the Internet.

Finally, we hope to combine our layout and historical data to create a movie of the growth of the Internet. It is not clear to us exactly how to do that, but it won't be possible at all without raw data. It is unfortunate we did not start collecting the data sooner.

3 Network Mapping

Our tracing data consists of paths from a test host towards a single host on a destination network. The list of possible destinations is obtained from the routing arbiter database[?]. This is a central registry of all assigned Internet addresses, including those privately used, perhaps hiding behind firewalls. Each provides a target network address, such as 135.104.0.0/16.

There are other databases we could use, and in retrospect, we probably should have. Other route ownership databases are available from MCI, CAnet, RIPE and ANS. We should also include any networks not contained in these lists, but which is announced in the core routing tables. We miss approximately twenty percent of the networks. We will add these routes when we start the multiple-source mapping described below.

The network scanner randomly picks an IP number on each network that is likely to be in use. This selection is biased based on a survey of commonly-used IP addresses. Essentially, we are performing a slow host scan over time until we find a responsive host.

If the trace reaches a host on the target network, the address is saved for future traces. More than half the traces end with silence (due to an invalid address or firewall), or an ICMP error reporting failure.

This technique only records an outgoing packet path. The incoming path is often different; many Internet routes are asymmetric, as ISP interconnect agreements often divert traffic through different connections. We do not know of a reliable way to discover return packet paths, but some ideas are discussed in section ??.

The path may vary between traces, or even individual probes, depending on outages, redundant links, reconfigurations, etc. This means we may occasionally discover paths that don't exist. Imagine a packet to Germany that is either routed through the United Kingdom or France randomly, for example. As alternate packets travel through alternate paths, we will infer connections between the alternate paths that do not exist. We believe that load-balancing over large stretches of the path is rare. In terms of outages and routing changes, the number of routes changing over the time of our scan should be relatively small in general.

The target, date, path data, and path completion codes are recorded in a simple text file, whose format is described in Appendix ??. The database is manipulated with traditional UNIX text processing tools (like *sed*, *grep*, *awk*, *perl*, etc.) and some additional C programs.

Each day's database is compressed and stored permanently. Copies are available upon request. The latest Internet database is available daily[?]. The compressed database is about 10–20MB; we periodically strip out old paths in order to keep its size down (We take special snapshots of the database before we do this, however).

3.1 Mapping, Not Hacking

We do not want our tracing to be confused with hacking probes, so we must proceed gingerly. We probe with UDP packets addressed to high port numbers ranging from about 33,000 to 50,000. Most intrusion detection systems recognized these as *traceroute* packets, though our port range is larger

[1] The official road map of Ontario is the best exception we know to this last rule.

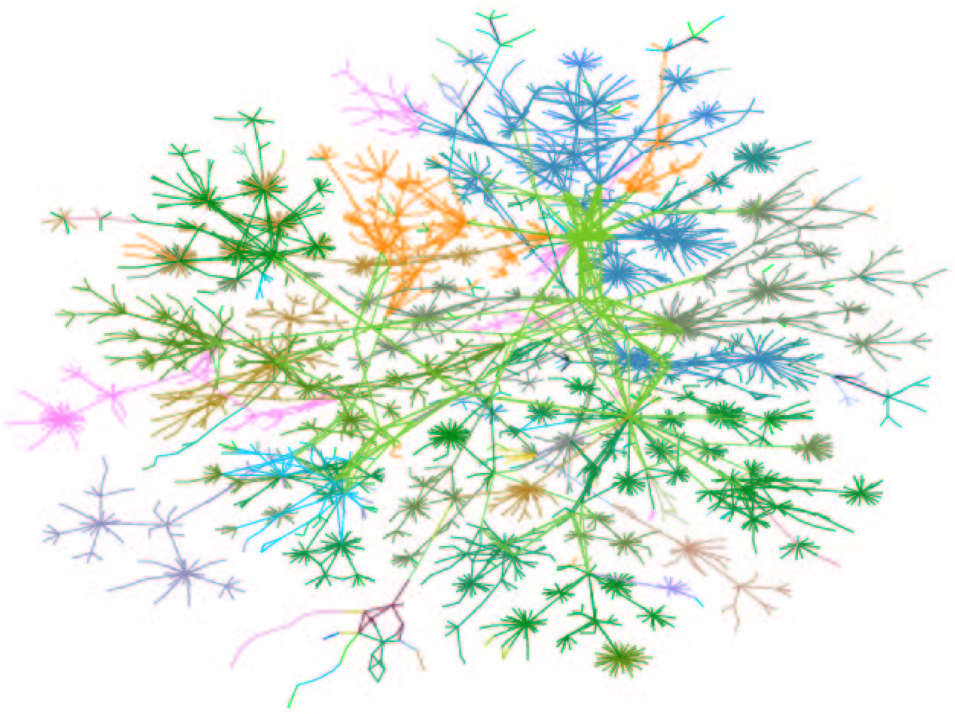

Figure 1: Lucent's intranet as of 1 October 1999. The map is colored by IP address, which tends to show communities within the company.

than *traceroute*'s. At worse, we tend to confuse system administrators; there are few real services that ever use these ports.

The path is discovered one hop at a time. If we get no reply for a hop, we try a second probe, then give up after a respectable timeout. This stops us from discovering the second half of many paths[?], but makes us a less threatening network citizen.

Since we do not want our mapping to be confused with hacking network probes, it is vital curious system administrators can easily determine what we are doing. The most important factor is probably the name of our mapping host, `ches-netmapper`, and the domain `research.bell-labs.com`. This name itself tells enough of the story to ease hacking fears, and we think this makes most administrators who do notice the packets nod and move on to other work.

We maintain a web page describing this project[?]. Tom Limoncelli, who has had to field a number of queries about our activities, added a DNS TXT record to `netmapper`'s entry which points to our web page. In addition, he suggested the world's shortest (and safest) web sever to direct queries to the project's web page.

A few network administrators have complained. They either did not like the probe, or our packets cluttered their logs (The Australian Parliament was the first on the list!). We record these networks in a whiner's list and cease probing them. Certainly others may have simply blocked our packets, or filtered our probes out of their logs. It would be interesting to compare hosts that were reached early in the scans and then later fell out of sight.

We have been in touch with a number of emergency response groups to explain our activity. We want them to understand the mapping activity and satisfy their justifiable curiosity. We would have a much harder time justifying our probes if we ran overt host or port scans, which often precedes a hacking attack. We believe only a tiny percentage of the Internet system administrators have noticed our mapping efforts.

The mapping machine itself is highly resistant to network invasion: some network scans have promoted powerful hacking responses. Of course, like any other publicly-accessible machine, it could fall to denial-of-service attacks.

4 Map Layout

We use a force-directed method to layout the algorithm similar to previous work[?]. The basic idea is to model the graph as a physical system and then to find the set of node positions that minimizes the total energy. The standard model which is employed is one of spring attraction and electrical repulsion. Attraction is done by connecting any two connected nodes by a spring. The repulsive force derives by giving each node a positive electrical charge, so that they repel each other.

Once you have this model, finding a minimum has been well studied. In particular, the most common techniques are steepest descent, hill climbing, gradient descent, and simulated annealing. We choose steepest descent because of the ease of coding it.

Previous work on graph drawing, however, has considered graphs the size of our Internet dataset as huge [?][?], and extending the runtime results of previous work to our graph and adjusting for a faster machine yield times on the order of months to millennia. Thus, the standard algorithms are too slow for our graph. We employ two tricks to more quickly compute a layout, at the cost of possibly being less optimal.

The first trick is replace the electrical repulsive field with spring repulsion. Any two nodes which are "close" to each other are connected by a spring whose rest length is how far we would like the nodes to be apart. This gives us a bounded repulsive force.

The real optimization, however, is laying out the graph one layer at a time. First the links to our three ISPs is laid out and the system is iterated until they stop moving "very much." Then, all the routers one hop further away are added, and the system is iterated (which may move the nodes from previous levels as well). Then the next hop, and so on. This tends to give placement based on information high in the tree. A movie of an early version of the layout process for Lucent data is available at [?].

Our original layouts showed all the paths. This resulted in a picture such as Figure ??. While the middle is mostly a muddle, the edges showed intriguing details. Note that a 36x40 inch plot is much more useful—a dense graph is easier to view on a larger printout. Dave Presotto described this smaller version as a smashed peacock on a windshield.

The map is colored by IP address—the first three octets translate into red, green, and blue. This simplistic coloring actually shows many communities and ISPs quite well.

We can already see features on this map. The fans at the edges show some interesting communities such as Finland, AOL, some DISA.MIL, and Telstra (Australia and New Zealand) all show up. The middle is very muddled, showing our ISPs at the time—Uunet (green) and BBN (deep blue). Sprintnet (sky blue) peeks through the sides.

The eye is drawn to the large purple and red splash, which represents the Cable and Wireless (cw.net) backbone, formerly the MCI backbone, formerly NSFnet. It is clearly the major feature (the magnetic north of the Internet) on the map. There are two reasons for this: (i) they are a huge backbone provider, and (ii) their backbone is an ATM network, connecting well over a hundred nodes around the world. Since our scanning is run at the IP level (level 3), this large network collapses to a single point. The smaller "Koosh" balls may be other ATM networks—we have not investigated this.

This map has changed over time, as we change our routing and ISP configurations. As we have done so, the predominant colors have changed as well.

We started collecting and preserving DNS names for the routers in March 1999. This data provides a rich source of data we can display using colors.

4.1 Spanning tree plots

Though posters-sized versions of this map were quite beautiful (and quite popular), they did not really meet our original visualization goals. The middle was a mess, and it did not look like we could iterate our way out of it, so the resulting map was not particularly useful.

When we computed and plotted a minimum distance spanning tree (which we will define as a spanning tree of the original graph such that the distance from the root is preserved), the picture became much clearer. This is a cheat in one sense—our packets do not always take the shortest path. But the clutter in the middle cleaned up nicely (see Figure ??).

If we consider only the shortest path to each destination, our graph turns into a tree, and the layout program can do a much cleaner job on it. Alternatively, we could have used a graphing algorithms designed to lay out trees of arbitrary size, which tend to be faster than the general algorithms.

Running our layout heuristic on the tree results in a very different map. The muddle in the middle is gone. The map looks less like a neuron and more like a coral fan or a space- filling curve. We can now trace individual paths

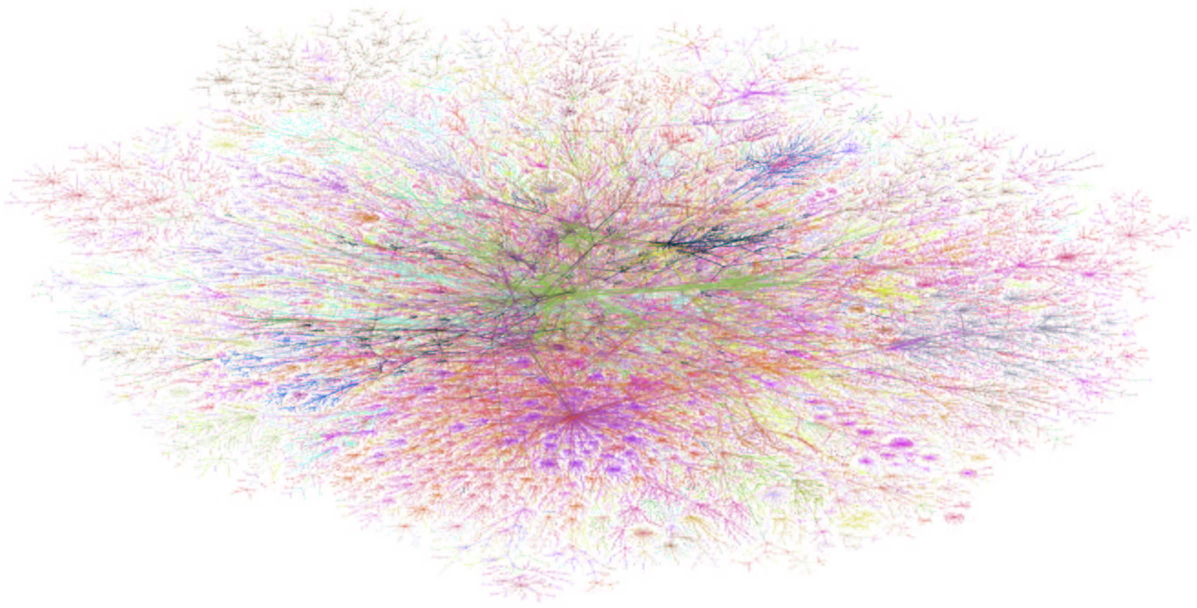

Figure 2: "Peacock-on-the-windshield" map from data taken in September 1998. This first appeared in Wired.[?] The blue and red star at the bottom is `cw.net`.

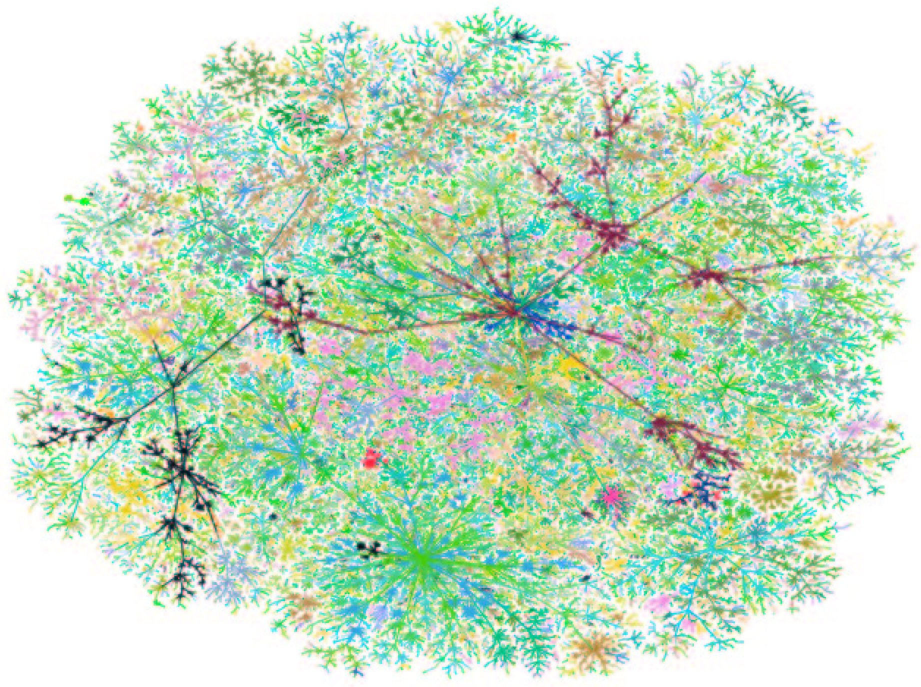

Figure 3: Minimum distance spanning tree for data collected on 2 November 1999. The blue-green star at the bottom is `cw.net`. The black foreground lines are links through net 12/8, Worldnet, one of our ISPs.

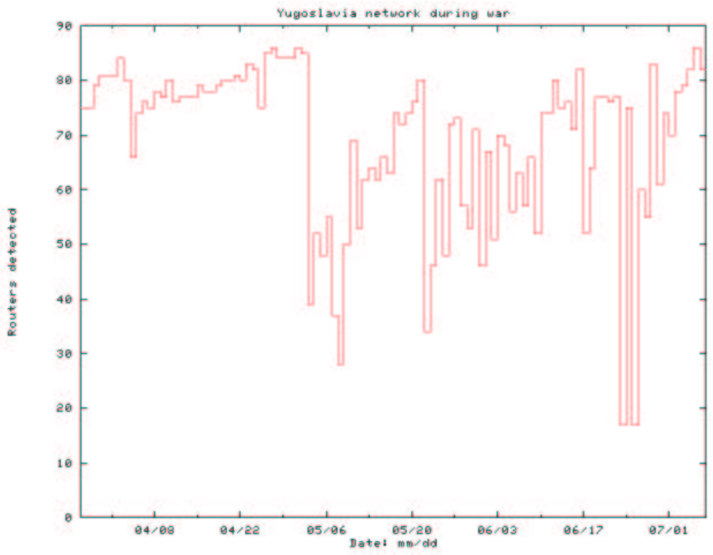

Figure 4: The number of reachable Yugoslavian routers over the course of the NATO bombing.

from our host to most destinations. The cw.net backbone is still spectacular, and still somewhat muddled.

We lose about 5% of our edges and 10% of our nodes when we throw away this inconvenient data. The edges still show interesting communities, but we can see much more now. By eliminating a number of inconvenient edges, we can make the map more useful, and traceable by the eye.

Now we can add those missing edges back in the background, in light cyan. In some cases, the alternate routes show up nicely. In others, the muddle is back, but out of harm's way. Some nodes attract a number of redundant connections, which the eye can pick out easily.

What works fairly well for the Internet works wonderfully for Lucent's intranet. That network has "only" 3,000 announced networks (versus some 90,000 for the Internet.) The full map is shown in Figure ??).

5 Yugoslavia

At Steve Bellovin's suggestion, we began doing a full trace daily to all the Yugoslavian networks we could find (down to the class C level) on 28 March 1999 (NATO's bombing began on 25 March 1999). Could we detect the effects of the bombing over the Internet? This question was inspired in part by efforts to determine the extent of the Loma Prieta earthquake using pings.

We scanned both Serbia (domain .yu) and used Bosnia/Herzegovina (domain .hr) as a "control" country. Did any Serbian internet links go through Bosnia? The answer was no; we never detected a cross- link between them.

We obtained the Yugoslavian target networks from various databases and double checked them with the routers in our own database. We split larger networks into class C networks (24 bit network address), and ran scans daily.

The topological data was interesting. Most of the links went to universities. The government's own network was even connected to a university router. Finding their propaganda pages were easy, even though the were in Serbian, as they had pictures of dead children and had words like *Nagasaki, Dresden, Baghdad,* and *Belgrade.*

(A quick port scan showed that this web site was not well protected. Suddenly, we needed a personal foreign policy because this site could be taken down! We didn't do it, but the ease of finding the web page shows some of the power of having a good map.)

The daily data did not vary much until the third of May, when the power grid was bombed. The number of reachable hosts dropped significantly (see Figure ??) and recovered unsteadily over the following weeks. It is interesting to note that the Bosnian sites vanished as well. Serbia is more industrialized than Bosnia, and the latter gets much of its electrical power from the former.

The graph of reachable hosts reveals the number of hosts that were available on a given day. By closely inspecting the graph, we see that the drop in reachable routers from 2 May to 3 May is the first significant drop. We ran these networks through our layout program (which doesn't do as well as other algorithms for such small graphs). Figure ??) shows which routers disappeared. The number of connections to two important routers in the upper right have been significantly reduced. Apparently, these routers were not not directly damaged, but that many of the outlying routers disappeared, possibly through power loss. There appear to be only three links into Yugoslavia.

We detected the results of distant damage in an semi-automated way. If a target country had a more extensive Internet presence, one could automate the detection of war damage. This could result in determining not only the extent of the damage, but also the location of it. We doubt that we are the first to have thought of this method of obtaining war damage information.

6 Related Work

There are a number of Internet data collection and mapping projects underway. Some have been running for a number of years, such as John Quarterman's Matrix Information and Directory Services[?], which includes the Internet weather report. Martin Dodge has collected many representations of networks at Cybergeography[?].

k claffy and CAIDA are collecting a number of metrics from the Internet with their *skitter*[?]. They have mapped the mbone, and collected path data to major web sites. We choose to map to each known network, preferring to map to everything that exists, rather than everything that is used (i.e. the web servers). We'd like to discover every possible path, not just those in use.

Internet maps are often laid out on the globe or other physical map. The desire to map the Internet to geography is compelling, but it tends to end up with dense blobs of ink on North America, Europe, and other well-connected regions. However, connections to distant and more sparsely connected regions can be represented nicely, *c.f.* Quarterman's map of connections to South America.

The problem with this method is the well-connected areas remain thoroughly inked, without a prayer of tracing paths through it. One approach is to simplify the map, showing connections by AS rather than individual routers. This is akin to showing the interstates on one map, and then creating local maps for each state. However, the AS connectivity graph is, proportionally, more connected than the IP graph, so the graph is still not very legible.

We simplify the graph in two ways. We trace to a single host on an announced network. Thus, MIT and someone's single class C network might each appear as a single dot on our map. We have also tried displaying only a minimum path to each point. This minimum distance spanning tree gives a much more readable plot at the expense of throwing away data.

Interactive visualization tools can aid in navigate a database like ours. One can zoom, query, and browse at will. It is hard to see the entire net clearly on a screen, though because there are far too few pixels. However H3Viewer[?][?] is one tool that looks like a good start to such a tool.

7 Future Work

Clearly, we need to expand the number of test locations. If we select enough of these, we should be able to fill missing links that we can't see now because routing is asymmetric or the links we never use.

We originally thought that we would need to locate computers worldwide, or obtain volunteers to run our mapping. Jorg Nonnenmacher suggested that we might offer a screen saver that displays an updated network map, and will perform modest mapping chores from sites scattered all over the world when instructed from a central site.

Jorg's suggestion is seductive, but it would have to be engineered very carefully to avoid abuse. The real problem, however, is that the tracing packets are slightly noxious. It would be best if we could preserve the return address, so they always appear to come from ches-netmapper. This makes filtering and reporting easier for those who watch and care about these packets.

Others have suggested that we use loose source routing to guide the probe packets down the desired paths. Though some have reported some success with this approach, we have found that a large majority of the Internet either blocks IP packets with options, or at least refuses to process them (We could display these nodes on our map—an interesting visualization). There is also the possibility that such "slow-pathed" packets may end up being routed differently.

We intend to use IP tunneling to distribute probe packets. We need volunteers to add a simple tunnel to their router for us. Then we tunnel packets to their router, with return addresses of ches-netmapper. Packets would trace outward paths from each tunneling router, and the results neatly returned to us. Sensitive sites would see familiar packets, though they may come in over new links. Of course, the tunneling routers would see each packet twice. These wide scans would need a lot more packets, so we probably couldn't run them daily.

The resulting data ought enable us build a mesh that closely describes the core of the Internet. We are not yet sure how to plot it—the data surely will look like our "peacock" and will need reduction or interactive visualization tools.

There is also a tricky problem sewing this data together. Traceroutes going in two different directions through a router may result in the router reporting two different IP addresses. How do we determine that those different IP addresses belong to the same router? We have a variety of heuristics

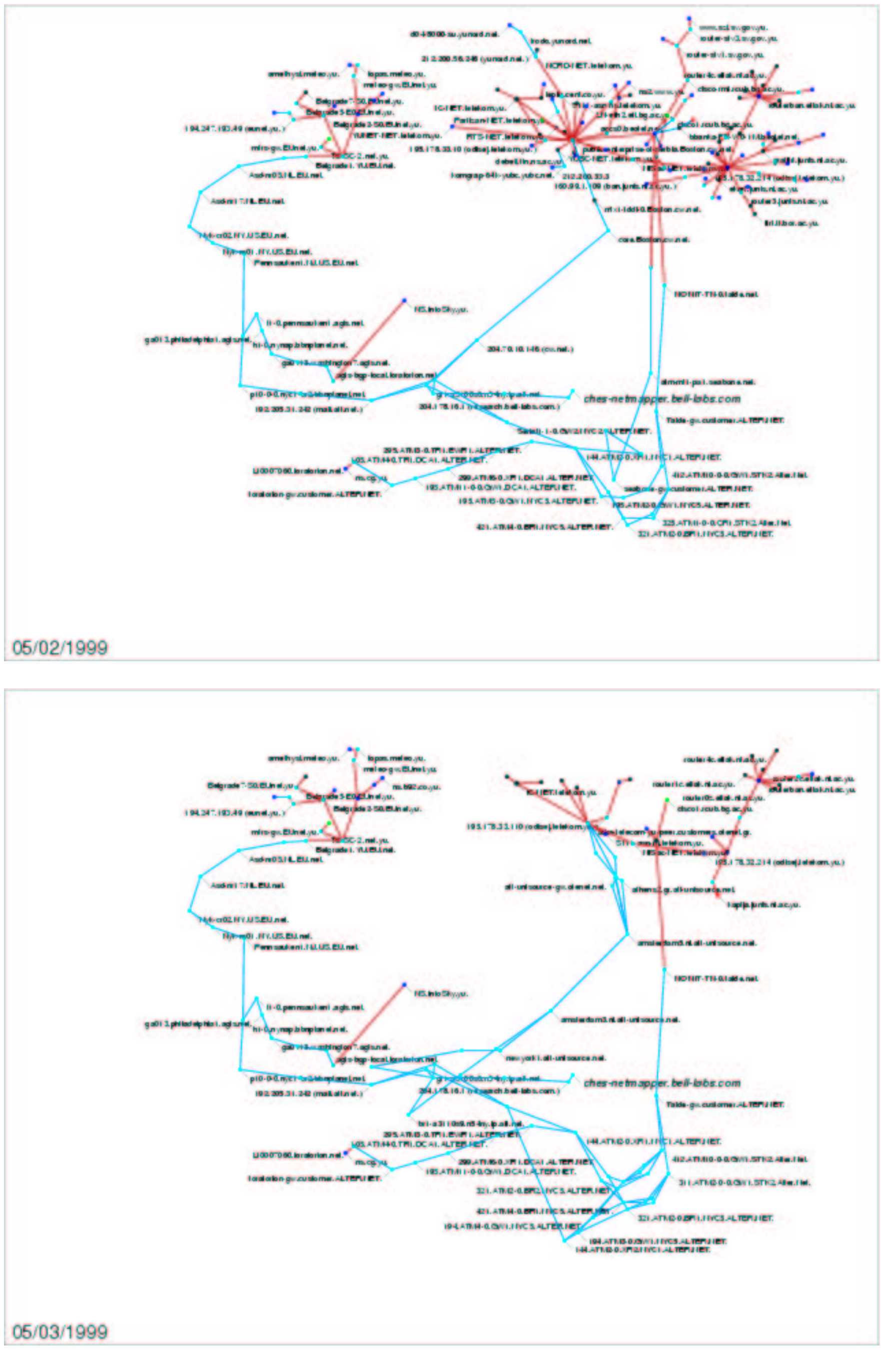

Figure 5: Map of Yugoslavia Internet connectivity on 2 and 3 May 1999.

we think will work most of the time, and hope that the graph theorists might be able to lend a hand.

We will still need to determine the number and position of sites needed to adequately map the “center” of the Internet.

Utilizing a third dimension in representing the graph is very tempting, either by doing the layout in three dimensions or using the third dimension to represent distance from us. However, the graph is too large for current VRML implementations that we are aware of, but ought to be easily handled by rendering engines. Of course, closer nodes may obscure distant ones.

Several people have taken our data to run through their visualization tools. Alas, modern displays simply lack the pixels to display the whole thing at once without some form of abbreviation. We look forward to their results.

We now have almost a year and a half of data concerning the Internet. We would like to create a movie of how the Internet's topology has changed over our dataset. The problem is making the picture for January 12th look enough like the picture for January 11th that the movie is fairly smooth while still showing a decent picture for both days. This is complicated by the fact that companies change ISPs, ISPs change internal connectivity, ISPs change peering arrangements, routing decisions, and ISPs change the IP addresses assigned to their routers.

8 Conclusion

The Internet maps have certainly caused a stir. They have excited the media, who is lacking for good visuals of the Internet (most Internet pictures show a web browser). The work has appeared in a number of publications[?][?][?][?] and is available as a poster [?].

However, there are more than just pretty pictures here. We have used our intranet maps to help show our company's connectivity. Some intranet maps clearly show routing leaks and other errors. We have used colors to show insecure regions, new acquisitions, and rare domains, which usually denote a leak or misconfiguration. The maps helped debug our corporate routing table, which contained route announcements for `lsu.edu` and the US Postal Service.

A number of researchers have picked up this data and run it through their visualization tools or run graph-theoretic analyses of it, and one paper (that we know of) has resulted so far[?].

9 Availability

Low resolution versions of various maps are available on-line.[?] High resolution versions are available commercially. Machine-readable high resolution maps are not available, and the mapping and layout code are proprietary. The authors will attempt to layout interesting data sets on request, though the programs are tuned for the Internet data and layouts of significantly different types of data have not been satisfactory so far.

10 Acknowledgments

Tamara Munzner, Stephen North, and Steve Eick have guided us into the world of visualization algorithms and tools. k claffy, Daniel McRobb, and the rest of the folks at CAIDA have helped us with mapping issues and ideas. Tom Limoncelli suggested the simple web server, and helped with Lucent and Internet routing issues.

Tom Limoncelli, Bob Flandrina, Paul Glick, and Dave Presotto have all had to field queries and complaints about this project, and we thank them for continuing good humor to do so.

References

[1] *Internet Tomography*, K. Claffy, et al. Nature, January 7, 1999.

[2] *A heuristic for graph drawing*, P. Eades. Congressus Numerantium, Vol. 42 p.149-160, 1984.

[3] *A fast adaptive layout algorithm for undirected graphs*, A. Frick, A. Ludwig and H. Mehldau. Proceedings of Graph Drawing '94, 1995.

[4] *JIGGLE: Java Interactive Graph Layout Environment* Daniel Tunkelang. Proceedings of Graph Drawing '98, 1998.

[5] *Tracing Anonymous Packets to Their Source by Selective Denial-of-Service Probes*, Hal Burch and Bill Cheswick. *Submitted.*

[6] *Mapping the Internet*, Hal Burch and Bill Cheswick. IEEE Computer, Vol. 32, No. 4, April 1999.

[7] `http://www.cs.bell-labs.com/ ~ches/map/index.html`

[8] `http://www.cs.bell-labs.com/ ~ches/map/db.gz`

[9] `http://www.cs.bell-labs.com /~ches/map/lucent.mpeg`

[10] *Intranets: Walking the Perimeter*, Steve Branigan and Bill Cheswick. *Submitted.*

[11] *H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space*, Tamara Munzner. Proceedings of the 1997 IEEE Symposium on Information Visualization, October 20-21 1997, pp 2-10, 1997.

[12] `http://graphics.stanford.edu / munzner/h3/`

[13] Van Jacobson, traceroute.

[14] `http://www.mids.org/`

[15] `http://www.caida.org/`

[16] `http://www.internetweather.com/`

[17] `http://www.cybergeography.org/`

[18] Wired, December 1998.

[19] Konr@d, April 1999.

[20] National Geographic, *Cartography*, Jan. 2000.

[21] New York Times, *Circuits*, Sept. ??, 1999.

[22] `http://www.peacockmaps.com/`

[23] `ftp://ftp.merit.edu/routing. arbiter/radb/dbase/`

[24] *k clay* , private communication.

[25] *Modeling Internet Topology*, W.R. Cheswick, J. Nonnenmacher, R.K. Sinha, K. Varadhan. Submitted to ACM Sigmetric 2000.

A Database format and details

Each day's run produces three files: the path database, an updated list of router names, and a log. Each is in text form, suitable for processing by traditional UNIX filters. All three files are archived for long-term reference.

The log contains the collection information, with some lines containing a Greenwich time stamp.

A.1 Path database

The path database contains one line per target network, and is divided into fields separated by white space. The first field is the target network, in a familiar form:

```
135.104.0.0/16
```

The filters assume that all four octets are present.

The remaining fields are in the form:

```
<name>=[<date>:]value
```

where `<date>` has the form `yyyymmdd`, suitable for sorting (although not Y10K compliant).

The field types are listed below. Only the first four appear in current databases—the rest are deprecated and have not been used since fall 1998. Some fields may appear more than once, representing data collected at different times. They are usually sorted by date.

Name	Date	Value	Description
Path	yes	see below	path to target
Probe	yes	(none)	date of last test
Target	yes	IP addr	host on target net
Whiner	yes	email addr	don't scan this net
Asnpath	no	unused	deprecated
Name	no	net owner	not used
Complete	no	(none)	deprecated
Pathdate	no	date	deprecated

The path field contains a comma-separated list of IP numbers, possibly followed by a completion code. If no code is present, the path reached the target. The other completion codes are:

?	same as !?	deprecated
!F	ICMP filtered	firewall encountered
!H	ICMP host unreach.	bad guess for the target
!N	ICMP net. unreach.	firewall or filtered
!R		routing loop, deprecated
!L		routing loop
!Z	incomplete	deprecated
!!	incomplete	deprecated
!?	incomplete	no response

A.2 Label database

The label database has one entry per line. Each entry has three fields separated by white space: an IP number, a label, and the date (as *yyyymmdd*) it was collected.

The label consists of a name as returned by a DNS PTR lookup. If a domain reported “no such domain,” the name of that domain is given in parenthesis. This gives some idea of who owns the IP address. If there is no answer, the label is the IP number enclosed in less-than/greater-than symbols: `<135.104.53.2>`.