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 have been collecting and recording routing paths from a test host to each of over 90,000 registered networks on the Internet 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 applied these tools to understanding its size and connectivity. We have also detecting the loss of power to routers in Yugoslavia as the result of NATO bombing.

A simulated spring-force algorithm lays out the graphs that results from these databases. 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 edges, the remaining graph can be made 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, location of firewalls, etc.

This paper expands and updates the description of the project given in [2].

1 Introduction

Network administrators have long used Van Jacobson's *traceroute* [15] 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. The average Internet path from our scanning host is 17 hops, and the radius is 25. The average path is slightly lower than most previous measurements, mostly due to the scanning host being on the edge of Lucent's network and Lucent having 3 ISPs where the scanning host is connected.

By obtaining a list of all announced networks on an internet, and discovering the path to each of these networks, we 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, the picture is a reachability graph, not a complete map.

In the course of developing and testing our mapping software, we quickly discovered 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 work on intranet perimeter discovery [1].

Each morning the mapping program scans two separate networks: Lucent's intranet and the Internet itself. On Lucent's intranet, the mapping program does run full scans daily. On the Internet, our daily scans cover about one tenth of the destinations, reaching each announced network about three times a month. The mapping program runs full scans of the Internet about once a month. The Internet data is published on a web page [23] and saved to CD-ROM. We plan to run these scans for years.

This scanning allows us to detect long-term routing and connectivity changes on the Internet. We are likely to miss the outage of a major backbone for a few hours, unless it happens while we are scanning. But a natural disaster, or major act of terrorism or war, may well show up.

Due to the magnitude of the resulting databases, a method of visualizing it is required. The eye can help us gain some understanding of the collected data. 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.

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 that are not neighbors. We shuffle these points for 20 hours on a 400MHz Pentium to obtain the maps shown in this paper. The maps of Lucent, on the other hand, take 20 minutes to an hour to layout, depending on whether all the links are shown or just a spanning tree. Sample graph sizes are:

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

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 [3].

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 the status of networks that are not directly connect to theirs. No one is responsible for watching the whole Internet. Of course, given its size, the entire Internet is difficult to watch. 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*-style 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-world 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 systematically collect data daily, building a consistent database that can be used to reconstruct routing on the Internet approximately for any day where mapping was done, at least the paths from our scanning host.

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 [1].

Our visualizations of the Internet itself have attracted wide media interest [25] [26]. 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.

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 [28]. This is a central registry of all assigned Internet addresses, including those used only privately. 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 networks announced in the core routing tables but not contained in these lists. Prelimanry analysis of these sources reveal that we miss approximately twenty percent of the networks. These omissions will be corrected when we start the multiple-source mapping described below.

We need to scan towards a particular host on the target network. It is not particularly important that the host actually be present. The network scanner randomly picks an IP number on each network that is likely to be in use. This random selection is biased based on a quick survey of commonly-used IP addresses (e.g., the most common last octet is 1 and lower numbers are more common). Essentially, we are performing a slow host scan over time until a responsive host is found.

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 7.

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

The technique employed only discovers the IP path. Each link along this path may not actually represent a physical link. For example, if an ISP is running their backbone over ATM, then each link represents a virtual circuit that may travel through many ATM nodes. Thus, depending on how the ATM network is connfigured, such an ISPs backbone may appear to be completely connected, even though it isn't physically true. From an IP standpoint, however, detecting this is extremely difficult.

The target, date, path data, and path completion codes are recorded in a simple text format, described in Appendix A. The database is manipulated with traditional UNIX text tools and some simple additional programs.

Each day's database is compressed and stored permanently. Copies are available upon request. The latest Internet database is available daily online[23]. The compressed database is about 10–20MB: we periodically strip out old paths in order to keep its size down (Special snapshots of the database are taken before this, however).

3.1 Mapping, Not Hacking

We do not want our tracing to be confused with hacking probes, so the mapping must proceed gingerly. The mapping program probes with UDP packets addressed to high port numbers ranging from about 33,000 to 50,000. Most intrusion detection systems recognize these as *traceroute*-style packets, though our port range is larger than *traceroute*'s. At worse, the probes tend to confuse system administrators: there are few real services that ever use these ports.

The path is discovered one hop at a time. For each hop, a probe is sent out. If no reply is received in 5 seconds, a second probe is sent. If no reply is received to the second probe in 15 more seconds, a third probe is sent. If no reply is received within 45 seconds after the third probe is sent, the path discovery is halted. Stopping a path trace after failing only one hop stops us from discovering the second half of many paths [6], but makes us a less threatening network citizens. A new scanner will try one hop beyond these IP "holes", giving us some idea of what we are missing.

Since we do not want our mapping to be confused with hacking network probes, it is vital that curious system administrators can easily determine what we are doing. Our first clue to them is the name of our mapping host, `ches-netmapper`, and the domain `research.bell-labs.com`. This name itself tells most of the story, 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 [22]. Tom Limoncelli, who runs the network that contains our mapping host, has had to field a number of queries about our activities, added a DNS TXT record to `netmapper`'s entry that points to our web page. In addition, he suggested the world's shortest (and safest) web server to direct queries to the project's web page (the web server just `cat`'s a file).

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 an opt-out 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 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 precede 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 other 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 similar to previous work [8] [7] to layout the graph. 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 employed is 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 gradient descent[13], conjugate gradient[14], and simulated annealing[13]. We choose gradient 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 [9] [16], 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. Imagine that any two nodes which do not share an edge are connected, via infinite strings, to a spring. Thus, if the nodes are further apart than the rest length of the spring, there is no force applied. If the nodes are are closer than the spring's rest length, the spring is compressed, and the nodes are pushed 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 are 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 web page [24].

Our original layouts showed all the paths. This resulted in a picture such as Figure 1. 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 using IP address; the first three octets of the IP number are used as the red, green, and blue color values respectively. 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: Finland, AOL, some DISA.MIL, and Telstra (Australia and New Zealand). The middle is very muddled, showing our ISPs at the time: UUnet (green) and BBN (deep blue). SprintNet (sky blue) peek 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. The collection of canonical names provides a rich source of data we can use to color the graph drawing. For example, colors can be selected based on top-level domain, showing the approximate country location of the hosts, or second-level domain, showing ownership of hosts.

4.1 Spanning tree plots

Though poster-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

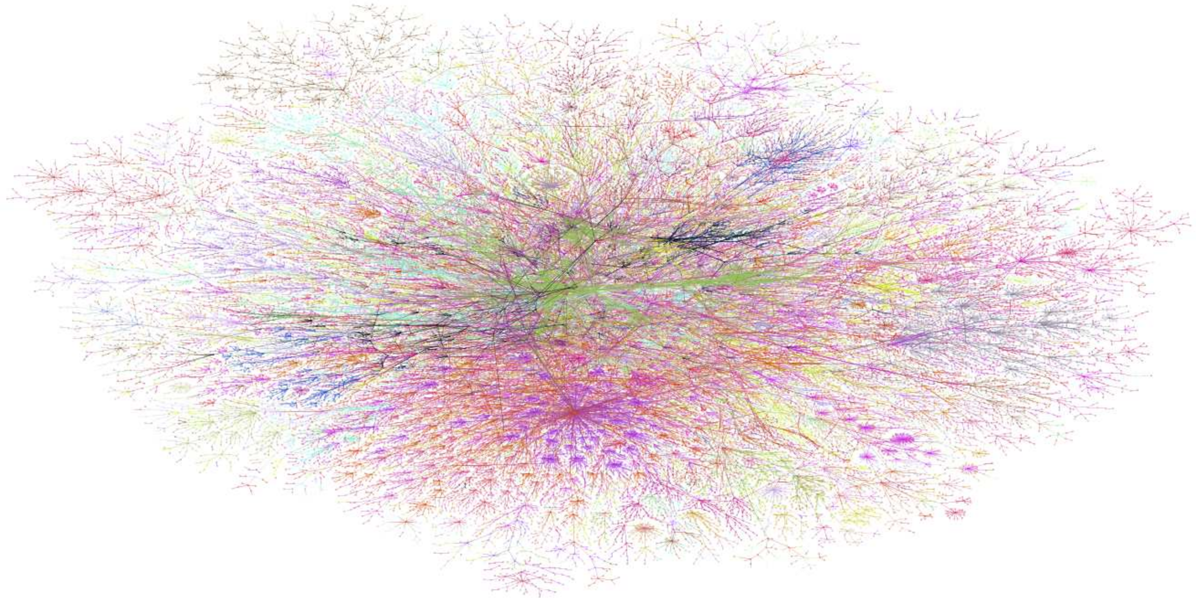

Figure 1: "Peacock-on-the-windshield" map from data taken in September 1998. The blue and red star at the bottom is `cw.net`.

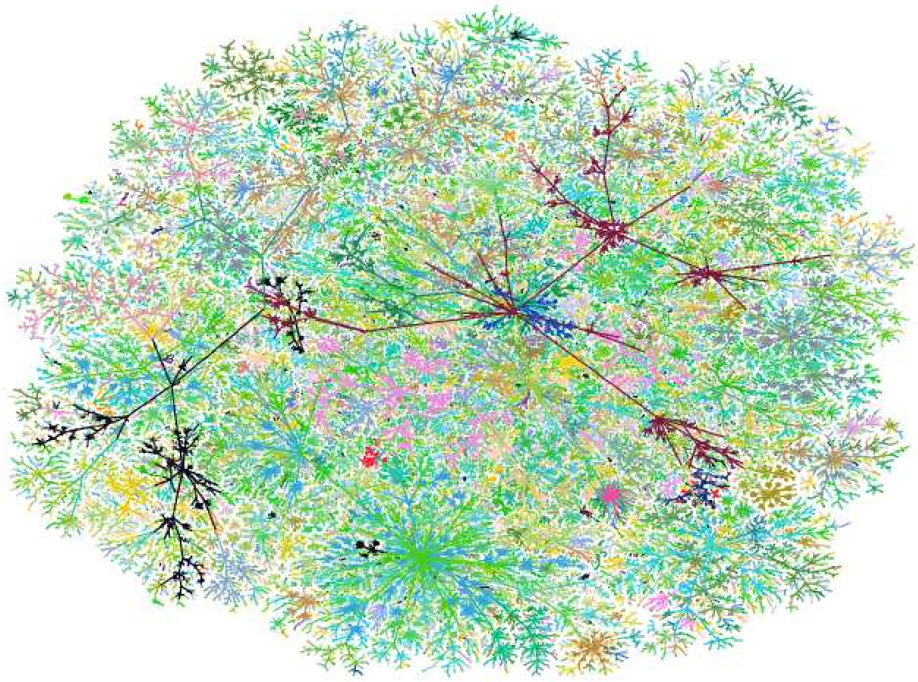

Figure 2: Minimum distance spanning tree for data collected on 2 November 1999. The blue-green star at the bottom is `cw.net`. The black foreground lines are links through net 12/8, Worldnet, one of our ISPs.

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 2).

If we consider only one 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 algorithm 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 from our host to most destinations. The `cw.net` backbone is still spectacular, and still somewhat muddled.

We lose about 5% of the edges of the graph 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 registered for the Internet at this writing.) The full map is shown in Figure 3).

5 Watching disaster

Internet monitors have detected major disconnections before; there were stories of ping utilities that incidentally mapped the extend of the Internet outage caused by the Loma Prieta earthquake using pings. Our data captured one aspect of NATO bombing of Yugoslavia in the spring of 1999.

During the first month of the war few if any Internet links were cut. But in early May, the bombing moved to the power grid, and the resulting disconnection is clearly shown in Figure 4. The connectivity returned slowly. Incidentally, the reachable routes in neighboring Bosnia also declined. We inferred (correctly) that Bosnia relies largely on the Yugoslavian power grid.

Figure 5 and Figure 6 compare days 39 and 40 of the NATO bombing. It is interesting to note two large spiny "Koosh" balls in the upper right of the map have been significantly reduced. This would seem to imply that although the core routers at the center of the "Koosh" balls were not directly damaged, many of the outlying routers were affected, possibly through power loss.

The maps also reveal that there appear to be only a few distinct routes into the Balkans from our test host. The power of the mapping technology is quickly apparent when viewing the limited number of gateways that appear, showing the connectivity of Yugoslavian domains with the rest of the Internet.

We detected the results of distant damage in an semi-automated way. We doubt that we are the first to consider the military uses. The usefulness is limited, because the exact physical location of most routers isn't known. Related techniques will doubtless be useful for monitoring the extent of other natural disasters, particularly in well-connected parts of the world.

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 [18], which includes the "Internet weather report." Martin Dodge has collected many representations of networks at Cybergeography [21]. An early attempt by Pansiot and Grad[12] mapped the path to 5,000 destinations. The Mercator project at USC [10] tries to get a picture of the Internet at a given instance in time.

In terms of long-term mapping, k claffy and CAIDA are collecting a number of metrics from the Internet with their *skitter* [19]. 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). Our goal is 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.

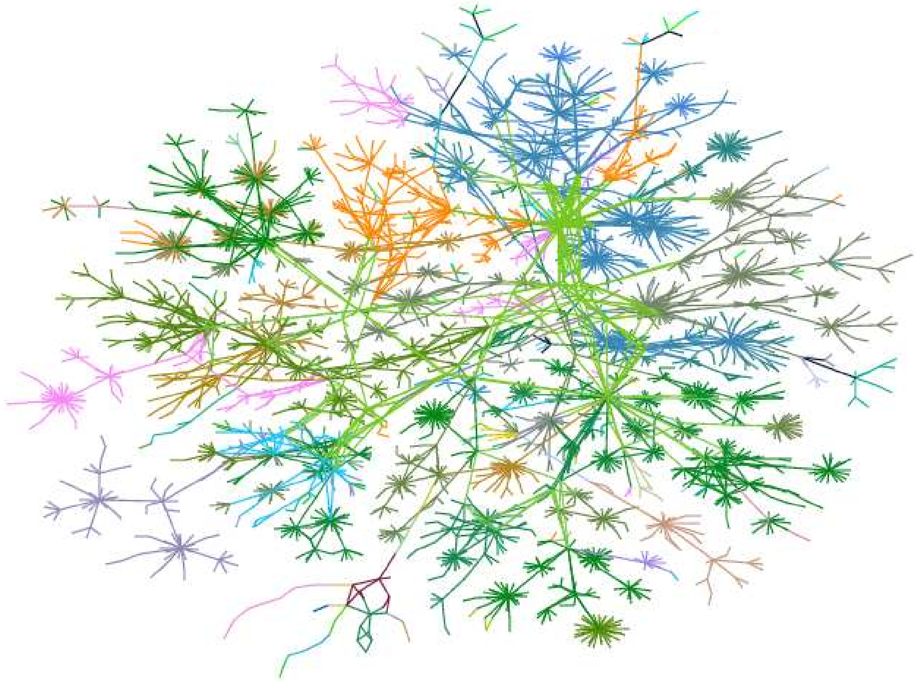

Figure 3: Lucent's intranet as of 1 October 1999.

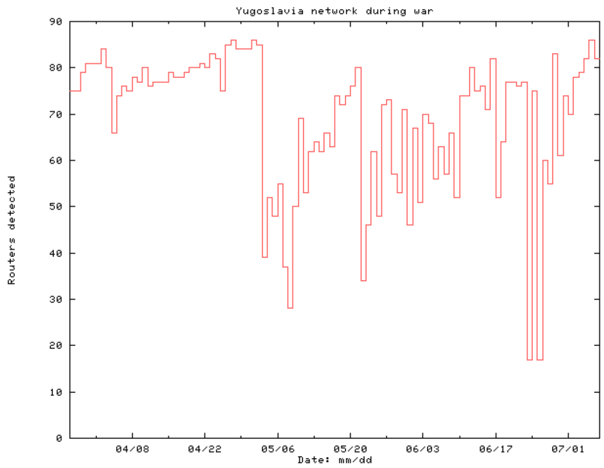

Figure 4: The number of reachable routers in the `.yu` domain over the course of the armed conflict.

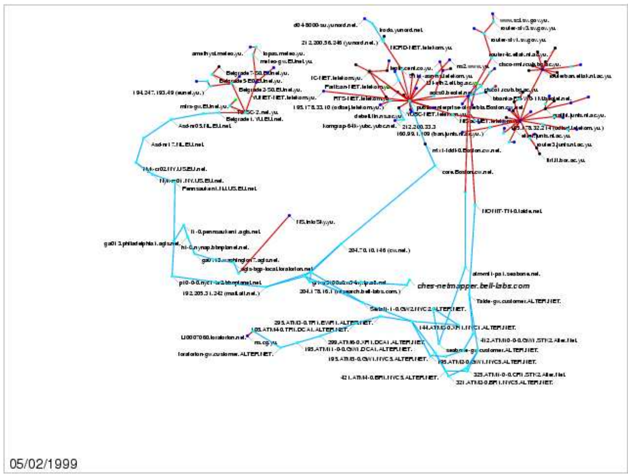

Figure 5: Map of paths to the Yugoslavian networks 39 of the armed conflict, colored by network.

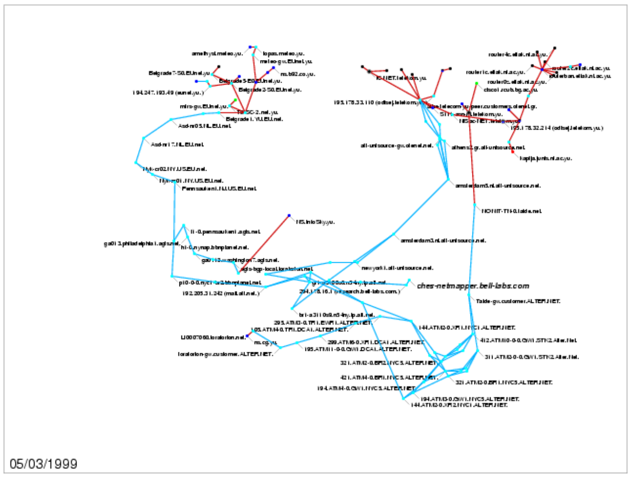

Figure 6: Map of the Yugoslavian Internet on day 40, colored by network. The main hubs in the upper right are still reachable, but they have lost a lot of leaves.

The problem with this method is the well-connected areas remain thoroughly inked, without a prayer of tracing paths through them. One approach is to simplify the map, showing connections by autonomous systems 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.

Interactive visualization tools can aid in navigating a database like ours. One can zoom, query, and browse at will. It is hard to see the entire net clearly on a screen: there are far too few pixels. However, H3Viewer [11] [17] is one tool that looks like a good start to such a tool. It displays a spanning tree of the graph and allows the user both to navigate the tree and also view the non-tree edges.

7 Future Work

At present, we are scanning out from a single test host. If we run the same scans from multiple hosts throughout the world, we will discover many more edges, and create a more accurate map of the "middle" of the Internet. We will discover the incoming paths to test hosts from the outgoing paths of other test hosts. 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 we never use them in out-going paths.

We originally thought that we would need to locate computers world-wide, or obtain volunteers to run our mapping. Jorg Nonnenmacher suggested that we might offer a screen saver that displays an updated network map, and would 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. And our layout tool only works on rooted trees at the moment.

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? There are several possibilities, including looking at the return IP of ICMP error messages [10].

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. The graph is too large for current VRML implementations that we are aware of, but ought to be easily handled by rendering engines. The other major problem is in order to avoid 'background clutter,' fog must be used, which means that a viewer only sees a local picture of the Internet at any given time.

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 two years 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 mapping technology can reveal insights about large networks. We've used these tools on intranets as well, 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 (domains with very few mapped hosts), 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.

The Internet maps, while seemingly less useful have certainly excited the media, who lacks good visuals of the Internet. [25] [26]. From a less scientific standpoint, the maps are interesting to look at, and one publisher created a poster out of it [27].

A number of researchers have picked up the routing database 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 [4]. As the data collection began in August, 1998, it provides a good deal of information about routing for a longer period of time than most routing studies to date have employed.

9 Availability

Low resolution versions of various maps are available on-line [22]. 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.

Our databases are also available at our web site, both the label database and the route database. Historic and current databases are available, along with the explanation of the database format from appendix A.

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 all have their names connected to the network that houses our test host, and have had to field queries and complaints about this project. We thank them for their continuing good humor to do so.

References

[1] Branigan, S. and Cheswick, W., "Intranets: Walking the Perimeter," *draft.*

[2] Burch, H. and Cheswick, W., "Mapping the Internet," IEEE Computer, Vol. 32, No. 4, April 1999.

[3] Burch, H. and Cheswick, W., "Tracing Anonymous Packets to Their Source by Selective Denial-of-Service Probes," *draft.*

[4] Cheswick, W., Nonnenmacher, J., Sinha, R., and Varadhan, K., "Modeling Internet Topology," Submitted to ACM Sigmetrics 2000.

[5] Claffy, K., et al, "Internet Tomography," Nature, January 7, 1999.

[6] claffy, k., private communication.

[7] Cohen, J., "Drawing graphs to convex proximity: an incremental arrangement method," ACM Transactions on Human-Computer Interaction 1997, v.4, no.3, p.197-229.

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

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

[10] Govindan, R. and Tangmunarunkit, H., "Heuristics for Internet Map Discover," Technical Report 99-717, Computer Science Department, University of Southern California.

[11] Munzner, T., "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space,"

Proceedings of the 1997 IEEE Symposium on Information Visualization, October 20-21 1997, pp 2-10, 1997.

[12] Pansiot, J.-J. and Grad, D., "On routes and multicast trees in the Internet," Proceedings of IEEE INFOCOM '97, Apr 1997.

[13] Russel, S. and Norvig, P., "Artificial Intelligence: A Modern Approch," p. 111-114, Prentice-Hall.

[14] Shewchuk, J., "An Introduction to the Conjugate Gradient Method without the Agonizing Pain," Carnegie Mellon University, School of Computer Science, unpublished paper

[15] Stevens, W. Richard, *TCP/IP Illustrated Volumn 1,* Addison Wesley, 1994, pps. 97–110.

[16] Tunkelange, D., "JIGGLE: Java Interactive Graph Layout Environment," Proceedings of Graph Drawing '98, 1998.

[17] `http://graphics.stanford.EDU/munzner/h3/`

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

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

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

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

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

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

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

[25] Wired, December 1998.

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

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

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

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, filtered, etc
`!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 nameserver reported “no such domain,” the domain of that nameserver 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>`.