



University of Twente

SURFmap: A network monitoring tool based
on the Google Maps API

Rick Hofstede, Tiago Fioreze
University of Twente, The Netherlands



Abstract

Network monitoring allows network managers to get a better insight in the network traffic transiting in a managed network. In order to make the tasks of a network manager easier, many network monitoring tools are made available for a wide range of purposes (e.g., traffic accounting, performance analysis, and so on) network managers may have. However, most of these tools lack to provide geographical information about network traffic. This paper presents a network monitoring tool prototype, called SURFmap, which provides network traffic information at a geographical dimension by using the Google Maps API. Through the use of the Google Maps API's features, SURFmap provides different zoom levels when showing network information, which results in the creation of different levels of abstraction in the network data visualization. SURFmap has revealed to be more intuitive when showing network traffic information, which makes the network monitoring activity from the network manager's perspective more interesting.



Introduction

- Network monitoring tools provide information about network traffic transiting within a network
- Available tools lack to provide geographical information about network traffic
- We propose a monitoring tool that allows network managers to visualize network information by using the Google Maps API's geographical capabilities

1. Introduction

Computer networks are complex communication systems that enable intensive interactions among users and services. Such interactions result in network traffic that should be monitored to check the health of the underlying communication infrastructure. Network monitoring, in addition, provides essential data to other network management processes, such as network optimization, accounting, and security. As a result, network monitoring turns out to be a critical task in any serious network management solution.

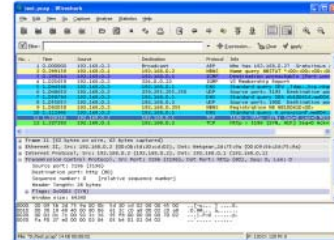
Network monitoring tools help network managers (or other end users) to handle network information. They usually provide some kind of user interface that presents network information. Nowadays, many network monitoring tools are made available, each one of them having a particular purpose and highlighting different aspects of network information. Some network monitoring tools focus on the inspection of packets, for example, whereas others focus rather on the monitoring of flows. Network managers often choose a particular network monitoring tool for a particular purpose. It is desired although that the chosen monitoring tool presents network information in a clear and easy way.

However, many of the current network monitoring tools lack to provide network traffic information at a geographical dimension. For some network manager's tasks, (e.g., attack tracking and user statistics) this feature could be desirable. Moreover, if a network manager would like, for instance, to get an overview of the geographical locations of network traffic passing by his/her managed network, he/she could get a simple but complete overview when a map, containing the geographical locations, is displayed.

In this paper, we propose a monitoring tool prototype that allows network managers to visualize network information through multiple levels of abstraction by using geographical capabilities provided by the Google Maps API [1]. The main contribution of this work is that it provides network traffic information in a geographical dimension.

Related work

- There are many network monitoring tools available:
 - Wireshark / Tcpdump
 - ntop / NfSen
 - and much more!!!
- Most of these tools lack to provide geographical information about network traffic
- The Google Maps API has potential to be used in the monitoring of network traffic



2. Related work

Most of the current network monitoring tools are developed for showing specific aspects of network traffic. The way in which network information should be presented can although be different for each purpose. Generally, most of the tools attend specific requirements on how and which network information is presented.

Wireshark [2] is a packet sniffer and protocol analyzer. It is able to capture and display all network traffic transiting at a particular network adapter. Furthermore, *Wireshark* recognizes the used protocols and is therefore able to put network information into its context. *Tcpdump* [3] is also a packet sniffer, but runs under a command line interface. The information provided by *Tcpdump* is basically the same as from *Wireshark*, but the latter provides a graphical user interface, whereas *Tcpdump* provides a textual one. However, both tools lack to provide any geographical information about network traffic.

At the level of flow information, monitoring tools such as *ntop* [4] and *NfSen* [5] are available. The main purposes of these tools are traffic measurement (measuring the usage of a particular network and maintaining statistics of that), traffic monitoring and network security issue detection. While both of these tools can provide a graphical user interface, they do not provide network traffic in a geographical dimension.

In the past few years, Google Maps has shown to have more potential than simply providing satellite imagery that allows users to drive directions in a fashion way. Google Maps has been widely used in several other areas [6] by means of the free usage of its API (Application Programming Interface). In the field of network monitoring, some research works have also been done. Van der Ham *et al.* [7] investigated the usage of the Google Maps API to visualize optical path finding in the GLIF infrastructure [8]. In another work, developed by Jamjoom *et al.* [9], the authors focus on the use of Google Maps' capabilities for the monitoring and management of network infrastructures.

The main drawback of the considered works is that they do not focus on the visualization of network traffic, even though some of them use the Google Maps API for their purposes. Since the Google Maps user interface has proven to be very intuitive by many different user groups, it could also be a very good starting-point as a new user interface for a network monitoring tool. Note that SURFmap is not meant to be a replacement for the current network monitoring tools, but to be an alternative!

Our approach

- To overcome the problem of many current network monitoring tools, we have developed SURFmap, which:
 - provides an user-interface based on the Google Maps API
 - adds a geographical dimension to network information
 - uses zoom levels to distinguish between various aspects of network information

3. Our approach

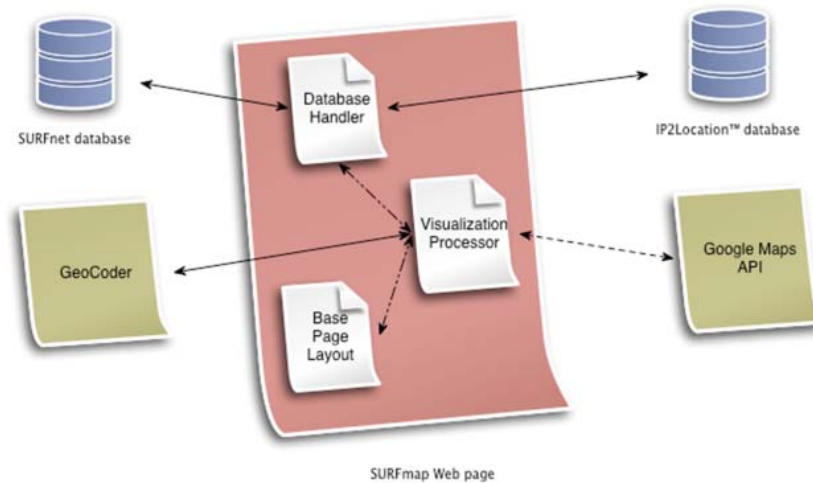
Since current network monitoring lack to provide geographical information about network traffic, we have developed a prototype tool that is intended to overcome this issue. Our tool, called SURFmap, provides a graphical user interface integrated with the Google Maps API. The latter has proven to be easy to use and provides map navigation functions in a fashion way. In addition, the zooming functions of the Google Maps API are implemented in SURFmap, in order to provide different levels of detail for the considered network information.

At its current development stage, SURFmap uses network information provided by SURFnet [10] from the Dutch research and education network, SURFnet6 [11]. This network information is collected from the SURFnet6 routers by using NetFlow [12] and correlated in a central storage database. SURFmap obtains the network information from this database and uses elements of the Google Maps API in order to visualize them.

In order to add a geographical dimension, SURFmap combines the network information obtained from SURFnet6 with the geographical information provided by IP2Location™ [13]. IP2Location™ provides several database sets to identify host's geographical locations (e.g., country, city, latitude, longitude, and so on). SURFmap divides the geographical information into several zoom levels and uses these zoom levels to show network information with different levels of detail.

Moreover, SURFmap provides four zoom levels: country zoom level, region zoom level, city zoom level and host zoom level of information. SURFmap enables its users to zoom in and out, through these zoom levels depending on the amount of details a user wants to obtain regarding the considered network traffic.

SURFmap architecture



3.1. SURFmap architecture

The system architecture of SURFmap is separated into multiple parts. The main part is the SURFmap Web page which is the core of our tool, since it manages all other external parts. This Web page consists of three subparts: the *Database Handler*, the *Visualization Processor* and the *Base Page Layout*.

The *Database Handler* is responsible for the communication with both databases used in the development of this prototype. One of them is called the 'SURFnet database', which contains all actual network data provided by SURFnet. The other database is called 'IP2Location™ database', which contains geographical information about IP addresses and IP ranges.

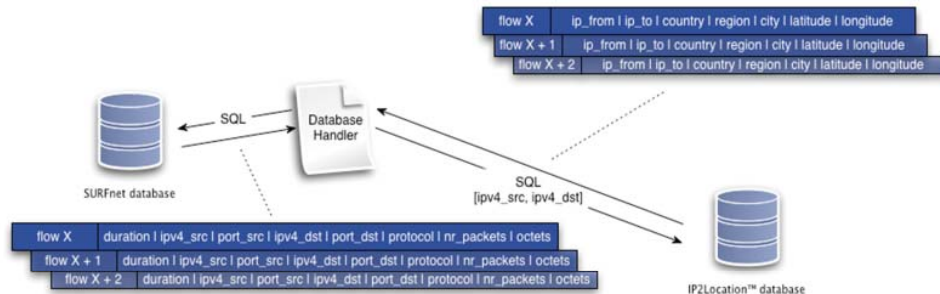
The *Visualization Processor* receives the network and geographical information from the *Database Handler*. After that, the *GeoCoder* is used to retrieve the coordinates (latitude and longitude) of the geographical information. SURFmap uses the *GeoCoder* provided by the Google Maps API. Another task of the *Visualization Processor* element is the creation of a map with its elements, using the Google Maps API; after formatting the network information, the Google Maps API's elements such as markers and lines are created using the geocoded information.

At last, the *Visualization Processor* sends the created map to the *Base Page Layout*, which puts it into a standard Web page with some extra elements, such as a help function and some navigation support.

The next subsections present detailed descriptions of the system parts presented above, as well as some application examples/screenshots.

Obtaining information from the databases

- Obtain network traffic information from SURFnet database
- Obtain geographical information about network information from IP2Location™ database



3.2. Obtaining information from the databases

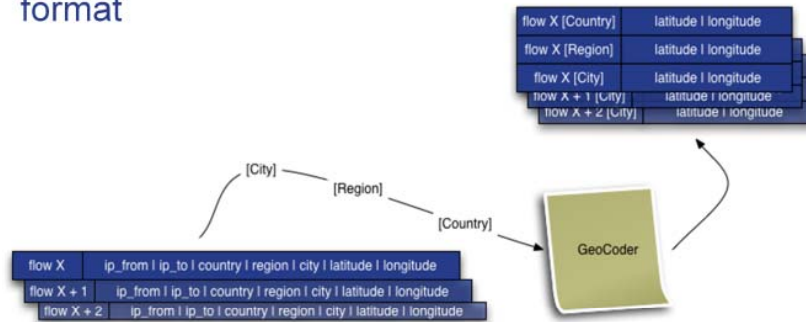
In order for SURFmap to show network traffic information, the *Database Handler* first needs to obtain the required information from the SURFnet and IP2Location™ databases. The SURFnet database contains the network information to be used by SURFmap, stored per flow and having the following fields: 'start_time', 'end_time', 'duration', 'ipv4_src', 'port_src', 'ipv4_dst', 'port_dst', 'protocol', 'packets' and 'octets'.

The first step performed by the *Database Handler* is obtaining network information from the SURFnet database by using SQL commands. The network information is then returned to the *Database Handler*, which takes all IPv4 addresses per flow (both source and destination addresses) and uses them to obtain the geographical information from the IP2Location™ database. Geographical information of both the sending and receiving host (in a flow) are obtained in this step. The information is stored in IPv4 address ranges and has the following fields: 'ip_from', 'ip_to', 'country', 'region', 'city', 'latitude' and 'longitude'. As the second step, the *Database Handler* obtains the geographical information about the network information collected in the first step. The IPv4 source and destination addresses obtained from the SURFnet database are used as input for the SQL query for the IP2Location™ database.

As an example, let us imagine one FTP communication from the host 1.2.3.4:21 ($1 \times (256 \times 256 \times 256) + 2 \times (256 \times 256) + 3 \times (256) + 4 = 16.909.060$) to the host 5.6.7.8:32657 ($5 \times (256 \times 256 \times 256) + 6 \times (256 \times 256) + 7 \times (256) + 8 = 184.551.176$) which lasted 1 hour (3.600.000 ms) and transferred 3MB (3.145.728 octets) of data within 1.000 packets. The *Database Handler* would first query the SURFnet database and obtain a row like this: "1; 3600000; 16909060; 21; 184551176; 32657; 6; 1000; 3145728". The source and destination IP addresses would then be used by the *Database Handler* to query the IP2Location™ database. The result could be, for example: "1; 1677216; 16909568; Brazil; Rio Grande do Sul; Porto Alegre; 30° 2'S; 51° 13'W" and "1; 83886080; 84281344; The Netherlands; Noord-Holland; Amsterdam; 52° 23'N; 4° 55'E". It is worth of saying that the IPv4 addresses are coded in the same way in both databases.

Processing information

- Geocode geographical information to get coordinates for each zoom level
- Convert network information into a proper display format



3.3. Processing information

The next step to be taken by SURFmap is the processing of information received from the *Database Handler*. Since the IP2Location™ database only provides the latitude and longitude coordinates of the last known zoom level, which is a limitation of the IP2Location™ database, the coordinates for the other levels have to be retrieved separately, using the *GeoCoder*.

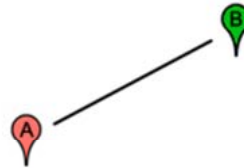
In order to display network information consistently, the information obtained from both the *GeoCoder* and the two databases have to be converted into some proper display format. For example, the IP2Location™ database returns “NEW YORK”, which is converted into “New York”. Similarly, the SURFnet database offers flow octets, which requires SURFmap to convert it into Megabytes (depending on the zoom level).

In general, the *GeoCoder* assigns geographic identifiers (i.e., geographical coordinates) to an arbitrary kind of data (e.g., country names). In order for the *GeoCoder* to process certain fields, such as the city name, SURFmap has to make an asynchronous call to the *GeoCoder* by providing that field name as an argument. The *GeoCoder* then returns the latitude and longitude coordinates for the supplied argument. At the end, the Google Maps API can use the provided latitude and longitude coordinates to place elements (e.g., markers) on the map.

As an example, let us imagine a host located at the University of Twente, which is located in Enschede, The Netherlands. The country name (provided by the IP2Location™ database) would then be ‘THE NETHERLANDS’, the region name ‘OVERIJSSSEL’ and the city name ‘ENSCHUDE’. To geocode this location, we have to make two calls to the *GeoCoder*: 1) for the country name and 2) for the region name (the city’s coordinates are already available, because that is the last known zoom level). The *GeoCoder* will then return two results, where the first contains the coordinates ‘52.132633’ and ‘6.89114’, which are the coordinates for ‘THE NETHERLANDS’ and the second contains the coordinates ‘52.436132’ and ‘6.42529’, which are the coordinates for ‘OVERIJSSSEL’.

Visualizing network information

- Plot data by using the Google Maps API
 - Markers provide information about end points
 - They show IPv4 addresses and their geographical location
 - Lines provide information about flows
 - They show information about the end points regarding the used ports, their geographical location, the exchanged amount of packets, octets and throughput



3.4. Visualizing network information

After all the information is obtained from the databases by the *Database Handler*, geocoded by the *Geocoder* and converted by the *Visualization Processor*, the *Visualization Processor* displays the processed network traffic information. Following the SURFmap workflow, the processed information available at this stage is the following:

- network information per flow;
- geographical information (country, region and city names) of end points per flow;
- coordinates (latitude and longitude) of the geographical information.

Since the Google Maps API has two main display components, markers and lines, we decided to make a clear distinction between them. Markers represent hosts and show information about them, such as their IPv4 addresses, the country, region and the city they belong to. On the other hand, lines represent a flow between two hosts (so between markers) and show information about that flow, such as the used ports at the flow's end points, the exchanged amount of packets, octets and throughput. The shown information is placed in information windows, which is an element provided by the Google Maps API. These information windows can be opened by clicking either a marker or a line.

Note that information about flows with the same end points are bundled together into one entry. As an example, two flows from 'The Netherlands' to 'United States' are written in the information window only once, but with the flow field counting as '2'. It is also worth of saying that SURFmap considers flows as unidirectional entries.

SURFmap zoom levels

- SURFmap provides 4 zoom levels:
 1. Country zoom level
 2. Region zoom level
 3. City zoom level
 4. Host zoom level
- Levels depend on information provided by IP2Location™ database
 - Unknown information is geocoded by using last known level of information

3.5. SURFmap zoom levels

As a consequence of the geographical dimension presented in SURFmap, our tool provides four zoom levels. The idea is that an application user can zoom in if he/she wants to get more information about some flow or host. The four zoom levels provided by SURFmap are:

- Country zoom level;
- Region zoom level;
- City zoom level;
- Host zoom level.

These levels depend on the information provided by the IP2Location™ database. As explained before, this database does not contain information about all fields for all IPv4 address ranges. In that case, the geographical information of the last known zoom level is used. From now on, we assume the country level to be the highest zoom level and the host level to be the lowest one.

The country level represents all countries which provide some network activity in the user-specified amount of flows. The same applies to regions, cities and hosts, so the region level represents all regions which provide network activity, the city level represents the cities and the host level the hosts in the user-specified amount of flows.

A user is able to zoom in or out using two different 'steps': 'zoom' is the ability to zoom in or out using the zoom steps provided by the Google Maps API. These steps are smaller than the steps provided by the 'quick zoom' feature of SURFmap. These 'quick zoom' steps are the steps mentioned above. Each of these steps consists of three zoom steps provided by the Google Maps API.



4. Application prototype

The screenshot above shows the initial page of SURFmap, which consists of the following elements:

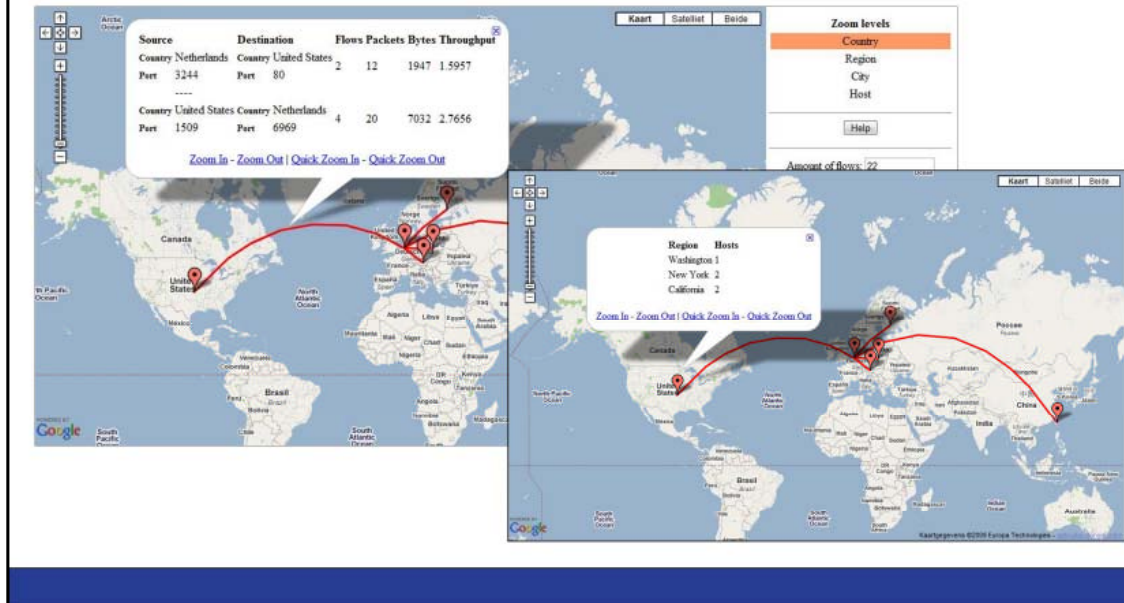
- a *world map*, provided by the Google Maps API. This is the main screen of SURFmap, that shows the visualization of the network information at the selected level of detail;
- a *table*, showing all available zoom levels and the current zoom level (marked red);
- a *help button*, which provides guidelines on how to use SURFmap;
- an *advanced mode button*, which enables a SURFmap user to edit more advanced SQL queries.

In order to start visualizing network information, the user selects the amount of flows to be displayed. The chosen amount of flows will then be selected from the SURFnet database following the order in which the flows were stored. Once the flows are selected, SURFmap retrieves the geographical information of these flows from the IP2Location database. At this stage, the user can start inspecting the flows information at their highest zoom level (i.e., country zoom level) or change to a different one. If the user decides to change the zoom level, the table to the right of the map will indicate the current zoom level. It is also possible to click at one of the zoom levels in the table to go to the selected zoom level directly. In short, the user is able to zoom in or out in three different ways:

1. using the zoom in and out links in the information windows;
2. using the table to the right of the map;
3. using the zoom 'slider' at the left top of the map, provided by the Google Maps API.

In the next subsections, we present three examples on the usage of SURFmap.

Zoom level grouping example



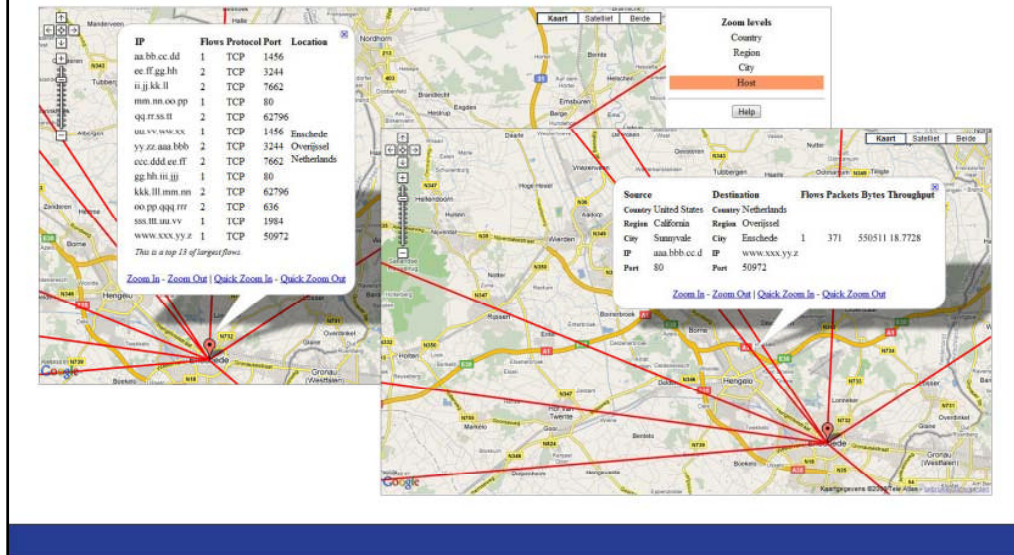
4.1. Zoom level grouping example

As mentioned before, SURFmap provides a zoom level grouping feature. This means that all active entities at a certain zoom level are fit together into one marker or line. An example of this can be found in the screenshots above.

SURFmap is showing that there is network activity between the United States and The Netherlands (leftmost screenshot). When clicking the red line, an information window pops up showing that there is communication from The Netherlands (NL) to the United States (US) and vice-versa. The window shows that there are two flows from The Netherlands to the United States, in which 12 packets with a total amount of 1.947 bytes are transferred. On the other way around, there are four flows from the United States to The Netherlands in which 20 packets with a total amount of 7.032 bytes were transferred. Even though the first two flows (NL → US) are in reality generated by different hosts, they are grouped into the same communication. Since the screenshot is taken at the country zoom level, zoom level grouping is based on the countries, in which the flow end points are located. The source countries of the first two flows are the same, just as the destination countries, which causes those two flows to be grouped together. The same counts for the last four flows from the United States to The Netherlands.

If the SURFmap user would like to know where in the United States the communication is coming from, he/she should click the marker representing the United States (rightmost screenshot). Since SURFmap is at the country zoom level, the informative balloon shows information of one level lower, that is, the region zoom level. Combining therefore the information provided by both screenshots, the user knows that the regions Washington, New York and California are communicating with The Netherlands, either as a source or as a destination.

Host zoom level example



4.2. Host zoom level example

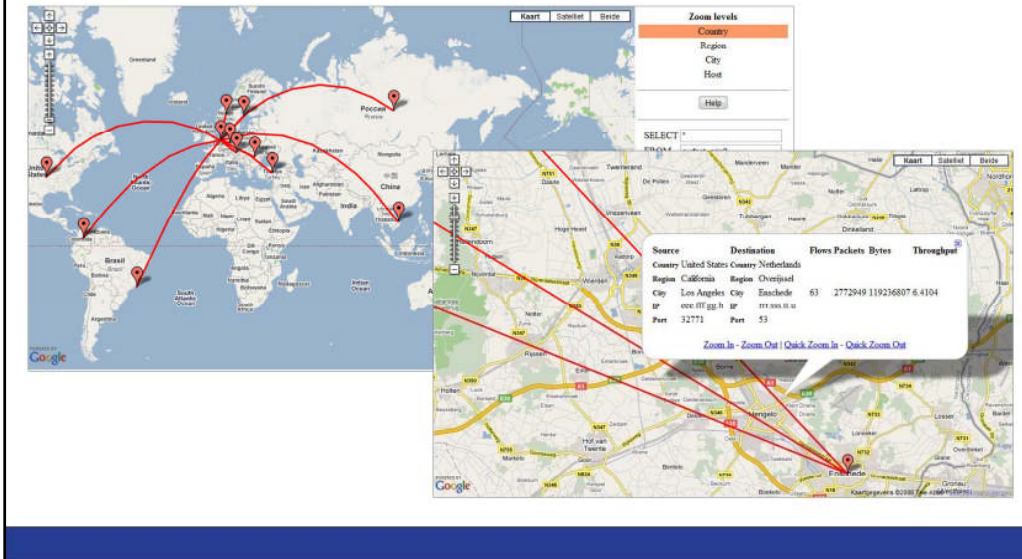
The screenshots above show the information provided by SURFmap at the host zoom level. Normally, the marker placed in Enschede shows all active hosts located in that city (leftmost screenshot). However, since there are more active hosts than do fit into the information window, only a subset of them is shown here. The information provided by the information window are the IPv4 addresses of the hosts in Enschede, the amount of flows to or from each host, the flow protocols, the ports at which the host communicates and the geographical location of the hosts. Since they are all in the same city (and thus in the same country and region) this geographical information is the same for all hosts in this information window.

The line's information window shows that there is one flow between the *aaa.bbb.cc.d** and *www.xxx.yy.z** hosts (rightmost screenshot). It also shows that the host *aaa.bbb.cc.d*, which is located in *Sunnyvale, California, United States*, exchanged 371 packets containing 550.511 bytes of data with the host *www.xxx.yy.z*, which is located in *Enschede, Overijssel, The Netherlands*. The throughput of this flow was 18,7728 KBps. Besides that, we can conclude that this flow belongs to HTTP traffic, since the source port of the flow is 80.

The next subsection presents a concrete scenario in which SURFmap appears to be very useful for network monitoring.

* Due to a non-disclosure agreement signed with SURFnet, real IP addresses cannot be shown. The real IP addresses were therefore replaced by fictional IPv4 addresses (e.g., *aaa.bbb.cc.d*).

Real-case example



4.3 Real-case example

In order to show the usability of SURFmap, we presented our prototype at SURFnet. SURFnet is a subsidiary of the SURF organisation, in which Dutch universities, universities for applied sciences and research centres collaborate nationally and internationally on innovative ICT (Information and Communication Technology) facilities. The presentation consisted of several cases, in which SURFmap appears to be very useful. One of them was a DNS attack on a University of Twente DNS server. The two screenshots above show this DNS attack.

The leftmost screenshot shows that several places were involved in the DNS attack. One of the attacker (or reflector) hosts was located in the United States. The rightmost screenshot shows that this host (having the *eee.fff.gg.h** IPv4 address) was located in *Los Angeles, California*. Besides that, SURFmap shows that there were 63 flows between this attacker host and the Dutch DNS server, in which 2.772.949 packets containing 119.236.807 bytes were exchanged. This means that the packets had an average size of 43 bytes.

After presenting our prototype tool, SURFmap gave us the following feedback to improve SURFmap: 1) let the line colors depend on the data volume of particular flows, and 2) integrate NfSen to let SURFmap show semi-realtime network traffic information every 5 minutes.

* Due to a non-disclosure agreement signed with SURFnet, real IP addresses cannot be shown. The real IP addresses were therefore replaced by fictional IPv4 addresses (e.g., *aaa.bbb.cc.d*).

Conclusions

- Current monitoring tools lack to provide geographical information about network traffic
- The network monitoring tool prototype presented in this work:
 - allows network managers to visualize network information using the features provided by the Google Maps API
 - adds a geographical dimension to network information

5. Conclusions

In this paper we presented a network monitoring tool prototype that shows network information using the Google Maps API. Our motivation for developing this tool was the fact that current monitoring tools lack to provide geographical information about network traffic. Since the Google Maps user interface seems to be popular in many different user groups, the development of a network monitoring tool using the Google Maps API could make network information easier to understand. Our monitoring tool prototype, SURFmap, has shown to be very intuitive when displaying network traffic information.

Following the recent trend of adding a geographical dimension to ordinary information (e.g., geolocation for vineyards), SURFmap adds a geographical dimension to network data. By adding this new dimension, SURFmap provides a totally different view on network information, compared to the information provided by current network monitoring tools. Moreover, besides using Google Maps API's elements such as markers to represent hosts, lines to represent flows and information windows to show information about either hosts or flows, SURFmap provides four different zoom levels (country, region, city, and host zoom levels) to more easily show network information about hosts and flows. By providing those zoom levels, SURFmap users can intuitively obtain more or less information, which makes it an important feature of our tool.

As future work, we intend to improve the way lines and markers are shown by SURFmap. Especially the animation of links, e.g. different line colors and thickness to make a clearer distinction between the various zoom levels and to indicate the network load, will be considered. In addition, we are also planning to integrate NfSen into SURFmap, because NfSen is a very popular network monitoring tool, which provides semi-realtime network traffic information every 5 minutes and offers the ability to filter this information quite fast.

Last but not least, we believe that the main contribution of this work is that SURFmap gives network managers a better, visualized network overview by providing network traffic information in a geographical dimension.

Thanks for your attention!

- Contact:
 - Rick Hofstede (r.j.hofstede@student.utwente.nl)
 - Tiago Fioreze (t.fioreze@utwente.nl)

References

1. Google. "Google Maps API", "<http://code.google.com/apis/maps/>". Accessed on January 12, 2009.
2. Combs, G. "The Ethereal Network Analyzer", "<http://www.wireshark.org>". Accessed on January 12, 2009.
3. Jacobson, V., Leres, C., McCanne, S. "Tcpdump", available at: "<ftp://ftp.ee.lbl.gov/>".
4. Deri, L., Suin, S., Carbone, R. "Ntop – Network Top", available at: "<http://www.ntop.org>".
5. "NfSen – Netflow Sensor", available at: "<http://nfsen.sourceforge.net>".
6. Google Maps Mania, "<http://googlemapsmania.blogspot.com/>". Accessed on January 12, 2009.
7. Van der Ham, J., Dijkstra, F., Grosso, P., Van der Pol, R., Toonk, A., De Laat, C. "A distributed topology information system for optical networks based on the semantic web". In: "Elsevier Journal on Optical Switching and Networking, Optical Switching and Networking", Volume 5, Issues 2-3, June 2008, Pages 85-93 "Advances in IP-Optical Networking for IP Quad-play Traffic and Services".
8. Global Lambda Integrated Facility (GLIF), "<http://www.glif.is/>". Accessed on January 12, 2009.
9. Jamjoom, H., Anerousis, N., Jennings, R., Saha, D. "Service Assurance Process Re-Engineering Using Location-aware Infrastructure Intelligence", Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on pp.439-448, May 21, 2007.
10. SURFnet. "Pioneering network", "<http://www.surfnet.nl/en/Pages/default.aspx>". Accessed on January 12, 2009.
11. SURFnet. "Map of the SURFnet network", "<http://www.surfnet.nl/en/netwerk/national/Pages/map.aspx>". Accessed on January 12, 2009.
12. Claise, B. "Cisco System NetFlow Services Export Version 9", Request for Comments: 2954 (RFC 3954), 2004.
13. IP2Location. "IP Address Geolocation", "<http://www.ip2location.com>". Accessed on January 12, 2009.