Mapping the African Internet

J.S. Gilmore, N.F. Huysamen, P. Cronje, M.C. de Klerk and A.E. Krzesinski
Department of Mathematical Sciences
University of Stellenbosch
7600 Stellenbosch, South Africa

Abstract—**This paper describes the methods used to generate router level maps and Autonomous System (AS) level maps of the African Internet and how these data sets are visualized. The *traceroute* utility was used to collect router level information on the Internet. AS level information was collected using a BGP feed connected to the South African Tertiary Education Network TENET. Router level maps were visualized using a geographical visualization and AS level maps were visualized using a three-dimensional hyperbolic visualization.**

We developed software to perform the following tasks: to automate the transmission of *traceroute* probes to selected IP addresses and to store and process the information produced by the *traceroute* utility; to transform the route data into graph data sets; to use the graph data sets, together with a geographical location database, to draw two- and three-dimensional router level maps of the African Internet; to process the BGP routing tables and create a graph of the AS level Internet; to translate the AS level graph into a data set that is rendered using three-dimensional hyperbolic visualization software.

I. INTRODUCTION

Previous work has been done to map the Internet both at the router level [1, 2, 3] and at the Autonomous System (AS) level [4, 5, 6]. Work has also been done on inferring AS relationships from router level maps [7].

This paper describes the methods that we used to generate a map of the African Internet. We map the Internet at the AS level, as well as at the router level, using several metaphors to visualize the data. As in [1] we used the *traceroute* utility [8, 9] to collect router level information on the Internet. We developed software to analyse the information produced by the *traceroute* data and to represent the data as an adjacency matrix. Public domain database software was used to store and process the *traceroute* data and the associated IP address and geolocation data. We used a BGP feed and an IP to AS mapping database to create an AS level graph of the Internet. Three methods of visualizing the graphs are presented, together with discussions on the information obtained from each visual metaphor.

The remainder of this paper is organised as follows. Section II describes the methods and software used to generate a router level map of the African Internet using the *traceroute* utility. Section III describes the methods used and the software used to generate an AS level map of the Internet, using both IP to AS mapping as well as BGP feeds. Section IV describes three methods for visualizing the data sets and discusses the characteristics of the different maps with focus on the African Internet. Section V discusses what was learned from the maps that were created. An initial version of Section II appeared in [10].

II. ROUTER LEVEL MAPPING METHODS

A. Database interface

The mapping software generates and processes large amounts of data. Data management must therefore be robust and queries to the data must be handled efficiently. The mapping software was therefore based on a public domain database implementation.

There are a number of databases available in the public domain, of which some well known ones are the SQL implementations. We used the SQLite [11] database. The characteristics of SQLite are:

- SQLite is a C library that implements a self-contained, embeddable SQL database engine.
- SQLite may be less configurable than MySQL, but it provides greater ease of use.
- SQLite incurs less overhead than MySQL, so the output file is smaller.
- SQLite is easily interfaced to C.
- SQLite can be operated in console mode, or via C functions that output the console commands to the database directly.

IP address data

Some of the functions that were implemented in the SQLite database are:

- Retrieve IP address ranges: This function returns the next unmarked IP address range from the IPRanges table. If the *traceroute* data were successfully written to the database, the IP range is marked as completed. When the retrieve function is called again, the next unmarked IP address range is returned.
- Write/Retrieve *traceroute* information: This function is invoked with a route structure as a parameter. The route is written to the database in the route table. All routes are written to the same table, and a destination IP address field is used to distinguish between the different routes.

Geographical data

The GeoLite City database by MaxMind [12] was used to retrieve the geographic location of router IP addresses. This is a public domain world-wide database that contains a list of

This work was supported by grant numbers 2054027 and 2677 from the South African National Research Foundation, Nokia-Siemens Networks and Telkom SA Limited.

IP ranges mapped to cities with their latitude and longitude shown as well.

This database is said to be 60% accurate to within a 25 mile radius in the USA and 50 to 55% accurate to within a 25 mile radius in South Africa. This means that there is a 60% chance that a specific IP address falls within a 25 mile radius of where the geolocation database reports it to be. The accuracy for the rest of Africa is not known. Other, more accurate, commercial geolocation software is available [12, 13, 14].

A function was written to query the GeoLite City database and retrieve the latitude and longitude for a given IP address. However, the GeoLite City database does not contain the latitude and longitude positions for all IP ranges. Queries to the database sometimes return latitudes and longitudes of zero for some IP addresses. These zero positions must be looked up manually. Other websites are queried for the information [15] and if no geographic information can be found for a specific IP address, the *whois* registrar information is used. Another database was set up to hold the zero position information and a program was written to update IP addresses with zero positions, that had earlier been looked up, and add location information from the zero position database. This decreased the percentage of zero position IP addresses from 10% to 2.5%

B. Discovering the routes

The *traceroute* utility sends probes to destination IP addresses, but the destinations themselves are largely irrelevant when *traceroute* is used to map the routers in the Internet. What matters is the route that the probes traverse to reach their destinations. This is because the intermediate routers are mapped, and not the destination hosts. Multiple probes are sent out, and the returned routes are used to build a connection graph of the Internet as described in Section II-C.

Some networks do not respond [4] to *traceroute* probes because many networks block the ICMP response packets from exiting their network. These packets are needed in order to discover the route. However, our experience is that most intermediate routers attach at least their IP address to the ICMP reply that is returned when a packet's time-to-live (TTL) expires. Some of these routers also attach their name. The fact that many routers attach their names is not always sufficient to reconstruct a route. If one router in the path does not reply with its name, the path is said to be broken. (See Section III C)

Design considerations

The *Tracer* program queries a database to retrieve an IP address to which it sends *traceroute* probes. The route information generated by the probe is received and stored in the database. *Tracer* is based on *traceroute-1.4a12-20* [9] developed by Van Jacobson at the University of California, Lawrence Berkeley Laboratories. The program was used in a number of mapping projects [4, 7]. It initiates a single *traceroute* to a target IP address.

General working

- *Tracer* is run by first specifying the country to map; *Tracer* queries the database for IP ranges from the specified country.
- Probes are sent to two IP addresses in the selected address range. These IP addresses are chosen at random out of the first and second half of the IP range respectively.
- As each TTL expired ICMP response is received, the information for that router is saved in a data structure.
- After the information for the whole route has been received, the route data structure is written to the route database.
- This process is repeated for every IP address range in the specified country, or in the whole database, depending on the mode *Tracer* is running in.

Modifications to the traceroute-1.4a12-20 program

- The code was made modular. This made it easier to expand the functionality of the program.
- The database interface functions described in Section II-A were added to the program.
- The program was modified to execute in a loop, continuously querying the database to obtain the next IP range.
- A data structure was implemented to store the route information sent back by the *traceroute* probes. The following route information is recorded: the router IP address, the router name, and the time taken by the ICMP packet to travel from the previous router to the current router. The time depends on the link medium and is also a measure of the distance between the two routers.
- The original program sent multiple probes to improve the chances of getting a response. If a probe was successful, all subsequent probes to the same target are unnecessary because the required information has already been collected. The program was therefore modified to send one probe and to send another probe only if no reply was received from the previous probe. The number of retries can be specified.
- The program closes all sockets and files and frees all its allocated memory.

C. Graphing the routes

After *Tracer* has collected all the data, a program is run over the entire database that reduces it's level of normality by adding the location of each router into a column in the same table as the main dataset returned by *Tracer*. This enables the *Terrapixate* program to run much more efficiently.

Once the update to the database is finished, the *Terrapixate* program is used to process these data into a format such that the *TerraPix* program can draw connection graphs. All the output is saved in a *SQLite* database that can be immediatly imported into *TerraPix*.

The output is in the form of a list of connections, where a connection consists of a source address, a destination address, the average latency between the two and the number of times this specific source / destination pair appeared during the *Tracer* run.

These connections are generated in the following way. The dataset returned by *Tracer* which contains the raw output data is traversed in an iterative fashion. With each iteration, the current line is queried for the router name, router address and location, as well as the next router name, address, location and latency. A check is then done to see both of these router addresses belong to the same destination address (to check that they are from the same trace). This pair is then checked against the existing list of connection pairs to see if a combination of these addresses already exist. If there is no existing pair, the connection is added to the list, otherwise the number of occurence variable for that specific pair is incremented.

This process is repeated for the entire database. In the event that no router name was returned, the following line will be used to pair up with the current router. This pair will then be marked as being broken. These links will then appear a different color to the rest on the *TerraPix* maps.

D. Future improvements

Multiple traceroute sources

Running *traceroute*s from multiple *traceroute* servers will improve mapping accuracy. When routing to a given destination, a defective router may be traversed, or more likely, a network/AS that restricts ICMP packet forwarding may be traversed. When routing from a different location, that network might be bypassed and the ICMP response packets might reach the sender.

With more than one vantage point, certain techniques can be used to identify IXP's as well as the multiple interfaces of routers [4].

The *Tracer* program might be improved by adding source routing capabilities. All probes can then be sent to a certain location first, and then from there to their respective destinations. This would give the same effect as having multiple vantage points. This method was successfully implemented in [1].

One drawback of this approach is that most routers have their source routing capabilities disabled in order to prevent abuse. However, some routers do have source routing enabled and not many vantage points are needed. If a few source routable routers can be found, it would greatly increase the accuracy of the mapping methods.

General Improvements to be made: More accurate commercial geolocation databases might be acquired to increase the accuracy of the geographical mapping.

Currently the method used to identify multiple interfaces also uses the fact that IP addresses that belong to the same router will have the same geographic location. When the IP addresses are mapped using the geolocation database, routers with the same positions are mapped to the same location. This automatically identifies the interfaces of routers. A more accurate geolocation database might therefore be able to distinguish between multiple routers.

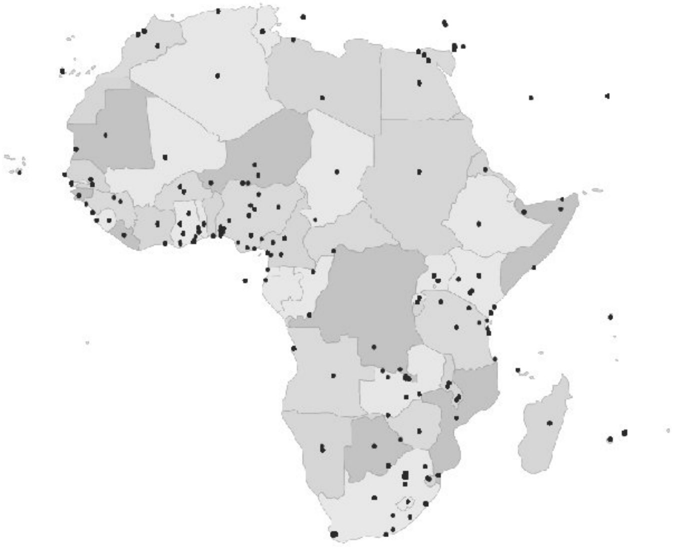

Fig. 1. A router level map of the African Internet showing only the routers

III. AS LEVEL MAPPING METHODS

A. BGP peers and routing tables

BGP information collected by the BGP speakers is required to map the connectivity of African Autonomous Systems (ASs). More specifically: information regarding the different paths available to IP prefixes, that are used by BGP, to perform Internet routing is required.

To retrieve the data, a BGP speaker was set up locally to peer with other ASs. This provides incremental updates to the routing information base (RIB), which may also be written to a file.

The RIB contains a table of all routing information known by the BGP speaker at a given time. This consists of a list of prefix/path pairs that lists the ASs, in sequence, that a packet should be routed through to a reach a given IP address. The updates may contain either route withdrawals by a peer AS, or single route additions to the routing table, for a given prefix.

These data files are input to the BGP data processor: the RIB first, followed by any updates. The AS-to-AS paths described in these data sets are then stored in RAM for processing (or removed, in case of a withdrawal update). These paths may be used to determine which ASs are connected to which, and in which direction they route traffic. A path may also indicate how many networks an AS is willing to route traffic for. Finally, this information is used to generate an adjacency matrix describing the connectivity of the ASs that may be used to generate a visual representation of the Internet. This visual representation shows how different networks in the Internet are connected.

B. IP to AS mapping

The MaxMind [12] database was used to map IP addresses to ASs in the African Internet. This database, along with the router level graphs are used to generate an AS level graph. The *Transcend* program was written to query the IP to AS database and convert all router names to AS names and then combine all ASs with the same name. This produced an adjacency matrix and names file that can be visualized as a graph structure.

This mapping method does not add any new data to the existing data set. It gives another perspective in which the data might be viewed, namely the peering relationships between different ASs.

C. Graphing the BGP routing tables

The BGP data processor maintains a persistent view of the entire network in the form of a list of prefix/path pairs, which is updated as it processes the data files. A distinction is needed between multiple prefixes carried via the same path. For example: in the event that an upstream AS decides to disallow routing for only one prefix out of five, the BGP data processor should remove only the single prefix link, instead of all five.

Whenever an adjacency matrix is generated, the prefix/path pairs list is transformed into a list of single AS-AS links, along with the number of prefixes routed over the link. The link list is sorted by the number of prefixes routed, followed by the AS number (ASN). This ensures that the nodes willing to route several prefixes to several destinations are listed first. The adjacency matrix that is generated, full or sparse, may then be used in a graphing program to generate the visual representation of the AS level graph.

RIBs contain *global* routing information, and so the BGP data processor also supports filtering the ASs included in the link list from an external file containing a list of permissible ASNs. This allows ASNs included in the output to be constrained to the African Internet segment. The list of permissible ASNs was generated from a list of ASNs in Africa, obtained from AfriNIC which is the Regional Internet Registry for Africa. This filtering process excludes any link which does not at least have one ASN listed in the AfriNIC list of ASNs. This limits the generated graph to African ASNs and their immediate non-African neighbours.

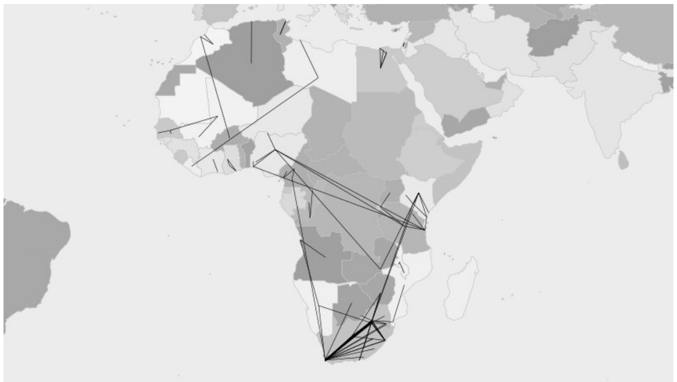

Fig. 2. A router level map of the African Internet showing intra-African routes only

Fig. 3. A router level map of the African Internet showing European routes

D. Future improvements

Currently the quality of the data is sufficient for the purposes of mapping the links between ASs since it *does* contain all the data that are available regarding the routing information. In terms of placement and labelling, the AS information contained in the filter file may be extended with location and naming information, extracted from the whois information for the AS.

Additional peers may provide a more complete view of the network: having only one peer would display only the routes leading away from it, revealing nothing about the links leading back from those other networks to the peer.

IV. VISUALISING THE AFRICAN INTERNET

Router level maps

Terrapix was used to draw the graphs. Router level maps are created using the *SQLite* database created from *tracer*. A world map was obtained on top of which the set of links was overlaid. This visual metaphor gives priority to node locations. All nodes may be easily viewed, while some links are obscured by others. Using this visualization, observations concerning node locations and connectivity by country or by city can be made.

Fig. 1 shows the positions of some of the major routers in the African Internet. Fig. 2 presents a map of intra-African routes. The map shows that routes that originate from the South African Tertiary Education Network do not make much use of the African infrastructure outside of South Africa. Figs. 3 and 4 show that many African routes are routed via the UK, Scandinavia and the USA.

When routing from a country in the south of Africa, for example South Africa, to a country in the north of Africa, for example Algeria, a satellite link to Amsterdam and then a link to Algeria are most likely cheaper (in terms of the IGP metric and BGP policy) than going through many fibre links in Africa. Although a satellite link has a high latency, it involves only one hop and in many cases will yield a two- or three-hop route to the destination. Many terrestrial links have low latencies, but will yield high hop count routes. Terrestrial routes thus appear to be more expensive end-to-end than satellite routes.

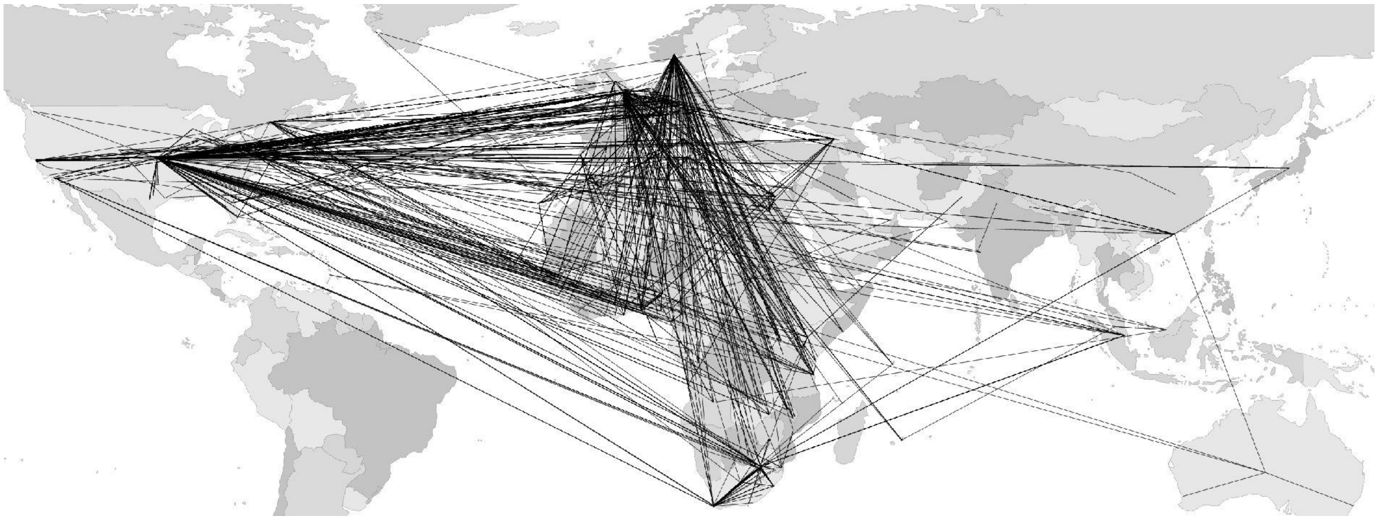

Fig. 4. A router level map of the African Internet showing global routes

This is however not the only reason for intra-African traffic being routed overseas. The first database that we created was for internal South African traffic. This map showed that internal South African traffic was being routed internationally, mainly via Washington DC, Hong Kong and Amsterdam. Once again, a satellite link is likely to be cheaper (in terms of the IGP metric and BGP policy) than going through several fibre links in South Africa.

The disconnected nature of the graph

To explain the disconnected nature of the graph, the way in which *Terrapix* constrains the graph must first be explained. In 2D mode (See Section IV-A) *Terrapix* allows the user to select a portion of the graph using the mouse. All nodes falling outside the selected region, together with their links, are discarded. This helps to reduce the clutter on the map.

When the Africa map is inspected, two apparent anomalies exist:

1) the map consists of a number of disconnected graphs
2) there are routers that appear to have no connections

There are several explanations for the disconnected nature of the African map:

- Packets routed to, for example, a destination in North Africa, are sent over international routers. When these packets enter Africa again, it is over an international link. These links are not shown in the constrained dataset. There is usually an international entry point where a packet enters Africa and from there the packet is routed to addresses in that area. This shows up on the map as a bloom effect, where we have a starting point, with links spreading outwards.
- Some graphs are near to each other, but are unconnected. There may be policy as well as logistical reasons why graphs that are geographically close to each other are connected.

  Another explanation lies in the way in which *traceroute* functions. The *traceroute*s that were done for Africa, sent probes to many IP addresses selected at random. Whilst many *traceroute* probes were sent to every allocated IP address range in Africa, it is not feasible to send *traceroute*s to every IP address in Africa. Whilst sending *traceroute*s to every IP address range should map most of the networks in Africa, IP addresses in a given range might not be near to each other. This can be due to a national company that has routers throughout the country or that runs many different networks throughout the country. This will cause that *traceroute* not to discover the other network and thus not discover the network router.

  Routing policies might also route packets coming from a certain area through a certain link. Therefore, there might be paths in the network that *Tracer* cannot find because of routing policy or broken links.
- Some routers do not append their information. It is then impossible to infer a connection from the first contiguous path segment to the second. For example, if the path `A-B-*-C-D` is returned, nothing can be said about how the route segment `A-B` is connected to the segment `C-D`.

The reasons why routers appear with no connections are due in part to the fact that *Terrapix* removed all their international links and *Tracer* could not find local links due to the reasons stated above. Another reason is that some routers do not append path information. When a path `A-*-B-*-C` is discovered, nothing can be said about the connections of `B`.

Link densities

The Internet entry points can be identified by examining the density of the outbound links at a specific point. The connections along the west coast of Africa are dense, because of the SAT3/WASC cable (South Atlantic 3/West African Submarine Cable) [18].

It is interesting to note the density of routers and links in the different African countries. South Africa has the most links.

South Africa	9,700,000
Morocco	590,000
Nigeria	302,000
Kenya	267,000
Algeria	152,000

TABLE I
IP ADDRESSES ALLOCATED TO AFRICAN COUNTRIES

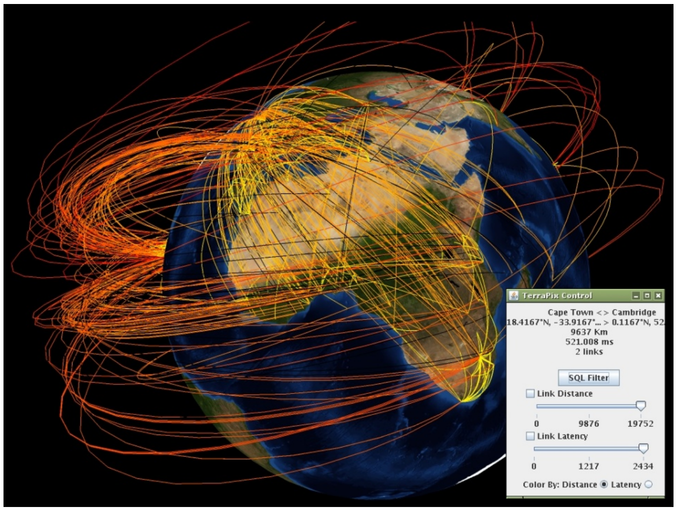

Fig. 5. A 3D router level map of the African Internet

This may be because all the *traceroute*s were sent from South Africa. The link density might look different if the *traceroute* probes had been sent from another country in Africa. On the other hand, significantly more IP addresses are allocated to South Africa than to any other country in Africa.

Table I shows the top 5 allocation of IP addresses in Africa. From this list it can be seen that Morocco with the second largest number of IP addresses has more that 16 times fewer IP addresses than South Africa.

It should be pointed out that these numbers refer to allocated IP addresses and not necessarily to IP addresses in use. There is no way of knowing how many IP addresses are in use at any given time in a country. IP addresses are allocated to companies or to Internet Service Providers (ISPs). Some of these IP addresses will be allocated to hosts for Internet access. The only way to know the number of IP addresses in any given range that are allocated to hosts, and the number of those hosts on-line, would be to *ping* every host in that range and check whether a response is received. This cannot be done instantaneously, and thus the picture that would be seen would be a time averaged one.

A. Two and Three dimensional geographic mapping

We developed the java based mapping program *Terrapix* to render the data in the *Tracer* database and produce 2D and 3D level router maps.

The first visualization method is two dimensional geographic mapping. World maps were obtained which were mapped to the the x, y plane. Nodes in the graph are then each given a position in the plane. The position of a node was chosen to be the physical location of the node, according to its latitude/longitude. Each Link was plotted as a line segment between two coordinates which were scaled to fit the given world map. The router level Internet can be visualized by overlaying the drawn graph onto a world map.

In 2D mode *Terrapix* produces and displays vector based (SVG) images using the Batik SVG tookit for Java [23]. This gives the abilty to zoom in to far greater detail than would have been the case had a simple bitmap output been produced. In addition the output may be exported for use in vector applications such as Inkscape or Adobe Illustrator.

Three dimensional geographic maps were created by mapping the world maps obtained to a sphere and plotting the series of links as arcs between geographical locations mapped to the points on the sphere. This mapping improves on two dimensional mapping, as discussed in SectionIV-A, by providing a greater amount of space in which to present the visualized data. In this visualization, routers are mapped onto a three dimensional globe, with links rendered as arcs.

The extra space is utilised by scaling the height of the link arcs as they increase in length. This creates multiple layers of links, where long links are coloured red. With these annotations, different types of links are easier to recognise and it is easier to draw meaningful conclusions from the map.

Geographical coordinates were read from the *SQLite* and converted to cartesian coordinates using simple geometry. From here piece-wise linear curves were created with adjustable accuracy by scaling a sine curve to fit between the two cartesian coordinates where the two connected routers lie.

Setting *Terrapix* in 3D mode activates the option to export to Google Earth (.kml file). When this file is opened in Google Earth the identical arcs to those drawn in *Terrapix* are drawn in Google Earth. Using Google Earth provides a vast amount of information about the pyhsical surroudings of a router, with the abilty to zoom to street level and see geographic features.

Terrapix offers the following features to help visualize the data

1) 3D mode allows for the height of arcs illustrating links to be adjusted so that links with a larger distance are scaled further into space.
2) Various world maps may be used including day, night and political maps. Night mode is particular interesting as the intensity of the cities lights are generally an indication of the population density with most links endpoints falling on such areas.
3) Filters are available allowing the user to display links on a slider according to their distance, density and latency.

The following data are represented in the maps:

1) The link end points are rendered onto the sphere at the locations of the corresponding routers.
2) The links show router connections.
3) Both the height and the colour of the links show the length of a link.
4) The link latency may also be displayed as the link

Fig. 6. Using Google Earth to display connectivity over western Africa

colour, by selecting the latency colouring scheme from the program menu.

5) The colouring scheme gradually changes from green, through orange, to red as the metric that it represents increases.
6) The thickness of a link shows how many interfaces connect the two cities (the endpoints of the link) on the map:

Observations

Fig. 5 shows a less cluttered view of the router level Internet. The high arcs from North-America to Europe represent traffic on the submarine cables connecting the continents. From this view, a large number of links may be seen going from Europe to Africa. These links originate from Oslo in Norway and Amsterdam in the Netherlands. There are large Internet exchanges in these countries.

Fig.6, Fig.7 and Fig.8 illustrate the advantage of using Google Earth which provides additional information about the physical surrounds of a router. Extra information include population density, unresolved city names etc. can also be obtained.

B. Three dimensional hyperbolic mapping

Router level maps are presented as geographic maps. These maps display the geographical positions of the countries and the physical locations of the routers in the network; the main purpose of such a mapping scheme is to show *where* infrastructure is located. To investigate *how* a graph is connected, a mapping scheme is needed that will give priority to the connectivity of the routers. It is difficult to present the structure of the AS level of the Internet using geographical mapping methods, because an AS often cannot be bound to one specific location. A graph visualization algorithm that shows the structure of a graph was used therefore to investigate the connection properties of the AS level Internet.

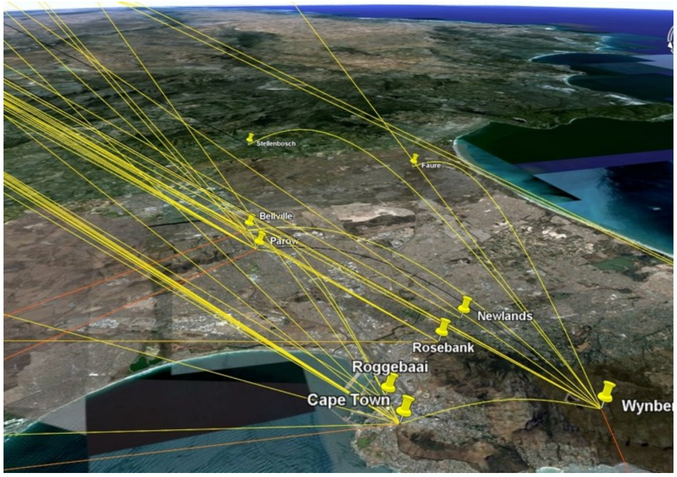

Fig. 7. City level view of Cape Town

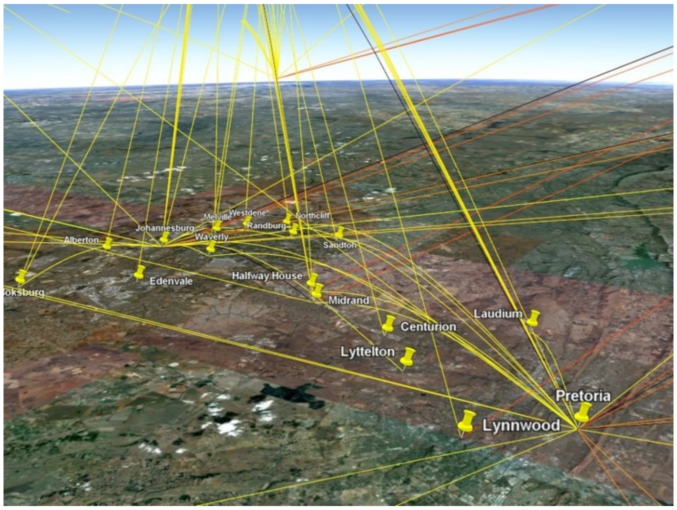

Fig. 8. Densly packet routers in Pretoria

Hyperbolic space

A three dimensional hyperbolic view was used as a visual metaphor to draw the AS level map of the Internet. The layout algorithm was developed by Tamara Munzner [19, 20, 21]. The 3D hyperbolic view is a fish-eye distortion where Euclidean points are mapped onto the interior of a sphere. The centre of the sphere is the origin of the coordinate system and the periphery of the sphere is at infinity. Objects effectively shrink as they go further away from the origin, creating a fish-eye distortion.

The drawing algorithm uses a minimal spanning tree (MST) representation of a graph being drawn. The root of the MST is chosen so as to highlight the tree structure of the graph as clearly as possible. If the graph has no discernible tree structure, the 3D hyperview algorithm should not be used. The mathematics of mapping hyperbolic space to Euclidean space (perceived space) are described in [22].

This visualization is useful when large graphs are viewed. A detailed view of the nodes in the area around the origin is obtained, while still being able to see the rest of the graph. From this viewpoint, a sense of how all the nodes fit into the bigger picture can be obtained. The location of the nodes in the graph do not correspond to the physical locations of the nodes. Every node's children are arranged around it in a hemisphere. Connection-based information can be obtained from this map with regards to the underlying structure of the Internet.

Generating the maps

The maps were generated using *Walrus*, a 3D hyperview program created by CAIDA [3] and based on the work done by Munzner. The program accepts an input file that contains a list of all the links in the graph, a list of the spanning tree links as well as a list of link and node attributes. A utility *Transmute* was written to read the adjacency matrix and name information of the router level graph and convert this information into the format used by *Walrus*.

The MST was computed using the router of the network where all *traceroute*s were issued from, as the root. When transmitting probes from a single location, a graph is created where the first routers detected are the routers from where *traceroute* traffic originates. These are the routers over which all trace traffic flow, and so the graph that is generated will have the first routers as the root of the graph.

The MST is computed using the first node specified in the connection list as the root. To create a graph with as few non-tree links as possible, the connection list is sorted from the highest degree of connection to the lowest. *Walrus* picks the first node as the root of the MST and so it will pick the highly connected node, generating the lowest number of non-tree links. It is important to note that the non-tree links are also part of the graph structure and should be included in the final representation of the network. The fact that there are non-tree links is an artifact of using a MST to draw the graph. Using a MST to draw the graph assumes that the graph being drawn has a tree-like structure which is not the case for the Internet.

Is the AS level graph disconnected? To create a MST, a fully connected graph is needed. All disconnected links are placed around the root of the graph to show that those graphs are disconnected.

AS level maps in hyperbolic space

Once BGP data had been obtained by peering with TENET, the data were processed by the BGP data processor and visualized by *Walrus*. In this section we discuss some of the observations that were noted while examining the resulting graph.

Fig. 9 shows an AS level graph of the global Internet, visualized in *Walrus*. Fig. 10 shows an AS level graph of the African Internet. The root of these graphs was chosen to be the highest connected router, which is "Internet Systems" and "TENET".

In Figs. 11 through 15, the blue nodes represent autonomous systems (ASs) that have their AS numbers (ASNs) registered by AfriNIC, while the red nodes are ASs that are registered elsewhere. The assumption was made that the set of AfriNIC ASNs represented the full African AS level space. Some entries were however manually added to the filter list: these entries were registered by RIPE, but were clearly African ASs.

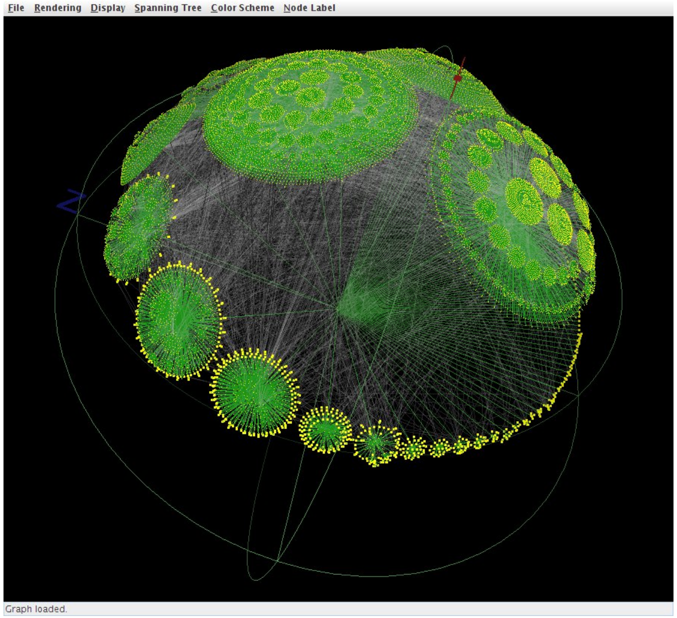

Fig. 9. A 3D hyperbolic view of the global AS level Internet

Figs. 11 through 15 were created by taking screen captures of *Walrus*, and then manually labelling relevant nodes by performing *whois* lookups on the ASNs. Some of the unrelated subtrees in the *Walrus* graphs were removed to avoid cluttering the graphs with irrelevant information.

The process followed when traversing the BGP graphs was

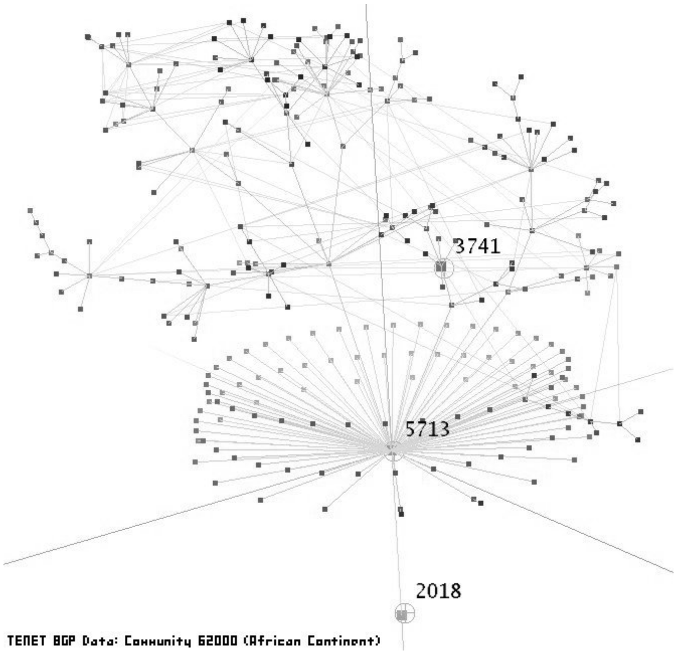

Fig. 10. A 3D hyperbolic view of the African AS level Internet

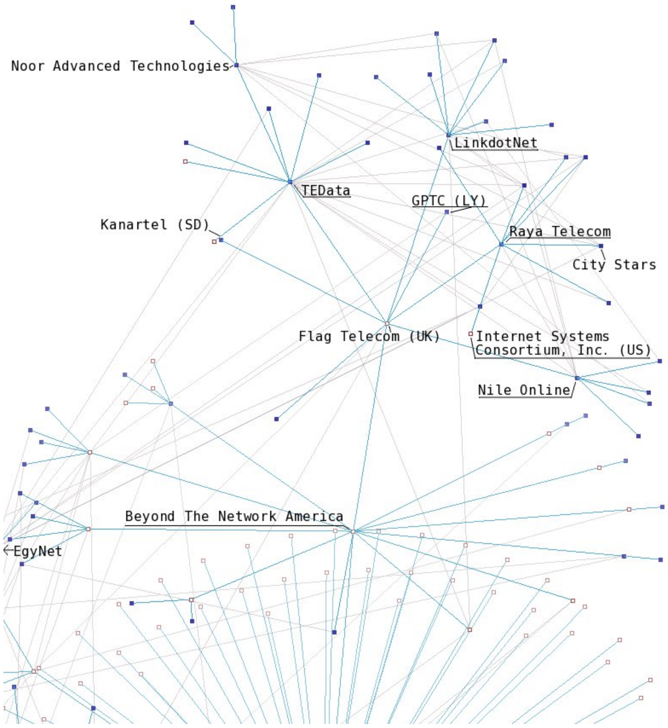

Fig. 11. Egyptian Autonomous Systems

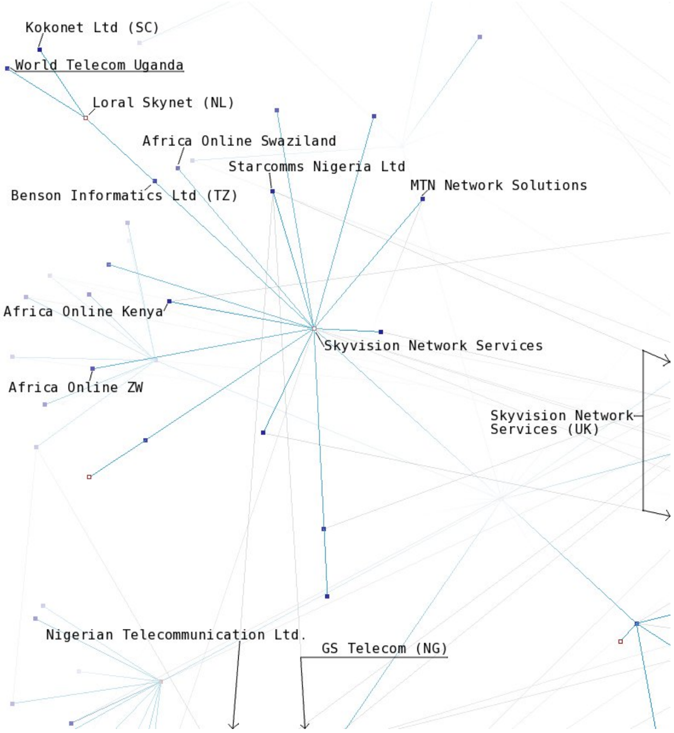

Fig. 12. AfricaOnline Autonomous Systems

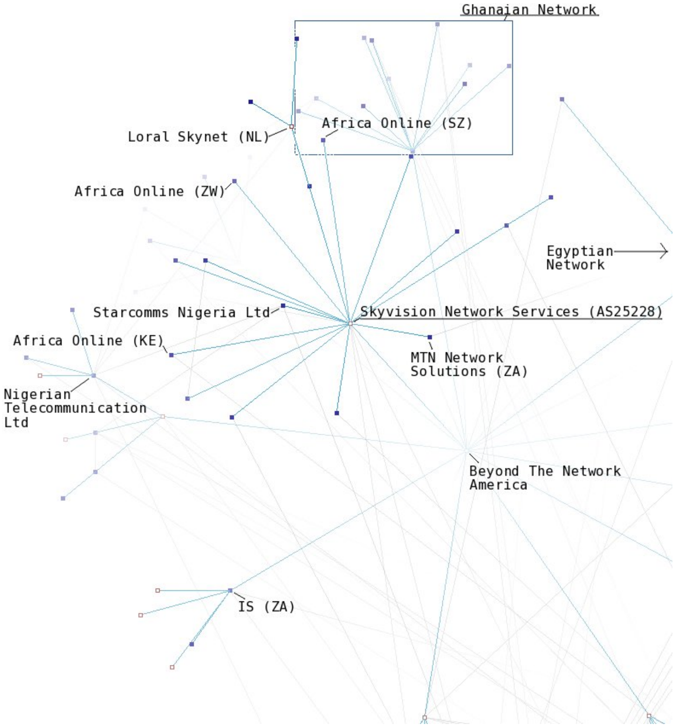

Fig. 13. Skyvision Network Services Subtree

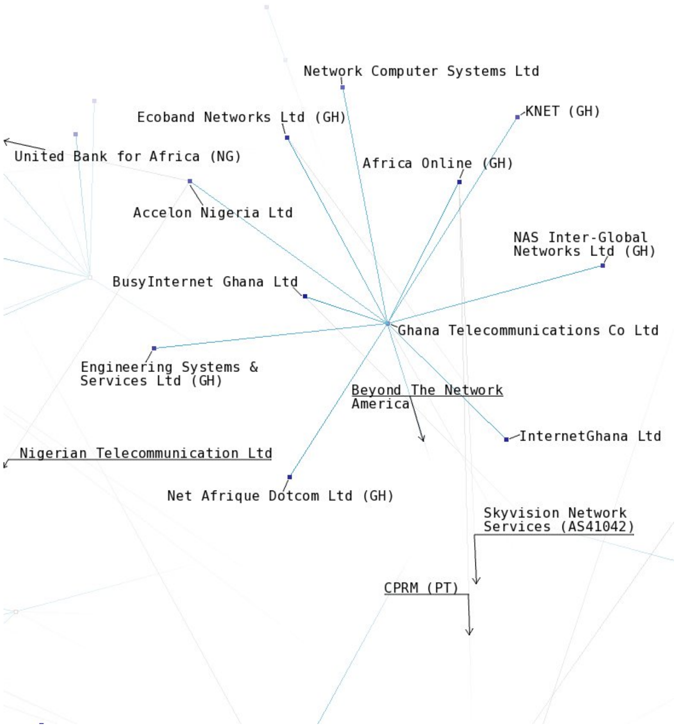

Fig. 14. Ghanaian Autonomous Systems

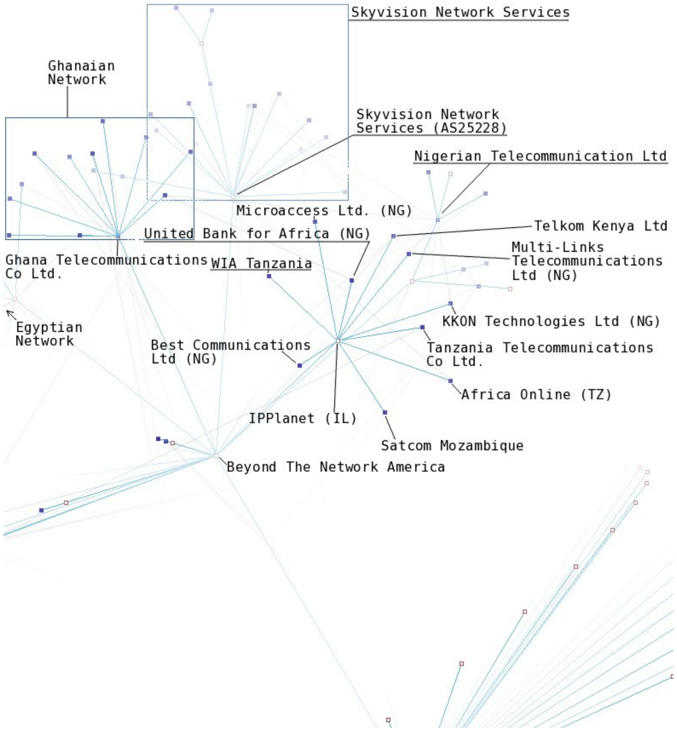

Fig. 15. Nigerian Autonomous Systems

to focus on a specific subtree and, through the use of whois lookups, map out which ASs peer with each other.

The first subtree examined, see Fig. 11, consists almost exclusively of ASs registered in Egypt. The root of the subtree was the Flag Telecom AS, which is an AS registered in the United Kingdom, which connects to a number of Egyptian ASs, with the exception of the Kanartel AS (Sudan), and GPTC (Libya). A number of links may be seen originating from the Egyptian ASs heading to another subtree off the left edge of the graph. These links are predominantly connected to the EgyNet AS.

Fig. 12 shows a subtree with the Skyvision Network Services AS at its root that connects a number of country-specific Africa Online ASs. For example Africa Online Zimbabwe, Africa Online Kenya, and Africa Online Swaziland are connected in this way. Some other ASs are also present in this subtree, such as the South African MTN Network Solutions AS. MTN's presence in this subtree is due to MTN's expansion into the rest of Africa. MTN has a prominent presence in the rest of Africa, outside of its presence in South Africa.

The same subtree in Fig. 12 is presented from a different perspective in Fig. 13, and shows the relation of the Skyvision Network Services subtree to some other subtrees that were mapped. It shows the parent of the Skyvision Network Services subtree as the Beyond The Network America (BTN) AS. It also shows how BTN connects to other subtrees, for example, the IS (South Africa) subtree, the Egyptian network given in Fig. 11, as well as a subtree that was identified as primarily consisting of Ghanaian ASs.

Fig. 14 shows the Ghanaian network referenced in Fig. 13, and illustrates how the subtree consists almost entirely of Ghanaian ASs. This subtree links via neighbouring subtrees to reach other countries. Fig. 15 shows another subtree of the BTN AS having a majority of Nigerian ASs.

BGP conclusions

The ASs seem to cluster in country-specific subtrees, containing only a small number of external nodes in the same subtree, as is shown by the Egyptian, Ghanaian and Nigerian subtrees presented in Figs. 11, 14 and 15. Connections to other countries (that is, subtrees) were either provided directly by links between the two subtrees, or indirectly via the subtree's connection to the parent (such as between the SkyVision Network Services subtree and the Ghanaian subtree, which is indirectly connected via their parent node, BTN). This concept of country-specific subtrees may also be extended to organisations, as is shown by the grouping of AfricaOnline ASs in the Skyvision Network Services subtree.

Another observation is that country-specific subtrees tend to have one larger AS (such as a country-wide telecommunications provider) that peers with international ASs, and acts as a bridge between the local leaf nodes (smaller national ASs) and the international ASs.

V. CONCLUSION

We present two methods of mapping the African Internet and three methods of visualizing the maps. These methods together present a picture of the African Internet. The AS structure of the African Internet was shown using the *Walrus* graphing package, as well as the geographical locations of its global and Africa components using the two and three dimensional visualizations.

The router level maps of the African Internet show the structure of the physical links and the AS level map gives insights into the relationships between the large ISPs in the Internet, and their peers.

REFERENCES

[1] R. Govindan and H. Tangmunarunkit. "Heuristics for Internet map discovery," in *Proceedings IEEE INFOCOM.,* 2000, vol. 3, pp. 1371–1380.
[2] N. Spring, R. Mahajan, and D. Wetherall. "Measuring ISP topologies with rocketfuel," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications,* August 19–23, 2002.
[3] Corporate Association for Internet Data, http://www.caida.org
[4] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. "Towards an accurate AS-level traceroute tool," *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications,* August 25–29, 2003.
[5] H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. "On Inferring AS-Level Connectivity from BGP Routing Tables," Proceedings IEEE Infocom (2002).
[6] B. Zhang, R. Liu, D. Massey, and L. Zhang. "Collecting the Internet AS-level topology," *SIGCOMM Comput. Commun. Rev. 35, 1,* Jan. 2005, pp. 53–61.
[7] H. Chang, S. Jamin, and W. Willinger. "Inferring AS-level Internet topology from router-level path traces," *Proceedings SPIE ITCom 2001, Denver, CO,* August 2001.
[8] G. Malkin. RFC 1393 Traceroute Using an IP Option, 1993. http://tools.ietf.org/html/rfc1393
[9] Van Jacobson, "Traceroute", ftp://ftp.ee.lbl.gov/traceroute.tar.gz

[10] JS Gilmore, N Huysamen and AE Krzesinski. Mapping the African Internet. *Proceedings Southern African Telecommunication Networks and Applications Conference (SATNAC)*, (Sept 2007), Mauritius.
[11] SQLite, http://www.sqlite.org
[12] GeoLite City database http://www.maxmind.com/app/geolitecity
[13] IP to location software from the company: IP2Location http://www.ip2location.com
[14] IP to location, NetAcuity software from Digital Envoy http://www.digital-element.net
[15] Geobytes, online geographic information tools http://www.geobytes.com
[16] Free vectorial world map by Fabio Visentin http://www.fabiovisentin.com/world_map/political_world_map.ai
[17] Google Earth, http://earth.google.com
[18] SAT-3/WASC/SAFE undersea cables http://www.safe-sat3.co.za
[19] T. Munzner and P. Burchard, "Visualizing the structure of the World Wide Web in 3D hyperbolic space," in *Proceedings of the First Symposium on Virtual Reality Modeling Languages,* 1995, pp. 33–38.
[20] T. Munzner, "H3: laying out large directed graphs in 3D hyperbolic space," in *infovis,* p. 2, 1997.
[21] T. Munzner, "Exploring Large Graphs in 3D Hyperbolic Space," in *IEEE Comput. Graph. Appl. 18, 4,* 1998, pp. 18–23.
[22] M. Phillips and C. Gunn, "Visualizing hyperbolic space: unusual uses of 4x4 matrices," in *Proceedings of the 1992 Symposium on Interactive 3D Graphics,* 1992, pp. 209–214.
[23] Batik SVG Toolkit for Java http://xmlgraphics.apache.org/batik/