UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2020

Title: Mr

Author: Blessed-Brighton Chitamba

Project Abbreviation: visAIT

Supervisor(s): Josiah Chavula

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	
System Development and Implementation	*0*	*20*	
Results, Findings and Conclusions	*10*	*20*	
Aim Formulation and Background Work	*10*	*15*	
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Visualizing Africa's Internet Topology

Blessed Chitamba
chtble001@myuct.ac.za
Department of Computer Science
University of Cape Town
Cape Town, South Africa

CCS CONCEPTS

• **Networks → Overlay and other logical network structures**.

KEYWORDS

AS, ISP Peering, IXP, Network Measurements,Internet Topology

1 ABSTRACT

This paper describes and analysed the process taken to recreate and visualize the internet topology of Africa as part of a project we undertook. The problem being addressed is that in the internet research space there lacks a unified platform with built in functionality to regularly collect internet measurements data, process it, and represent it on a map visualization with nodes representing Autonomous Systems and links representing the relationships between them. The key distinguishing feature about such a platform would be that it always reflects the current state of the internet topology at any point in time, according to the most recent measurements received. This would then allow researchers who intend to perform studies that involve the internet topology to carry out their research without having to perform their own internet measurement and topology visualization campaign, as the platform would be giving them the information they need. Hence, in this paper we describe the process we took to design and build an implementation of such a visualization and then analyse how useful it would be under different settings.

2 INTRODUCTION

The world now heavily relies on the internet, and it has become the biggest network of interconnected entities. Its continued growth brings new challenges and problems that increasingly deserve research attention. One of such problems is properly mapping and visualizing the topology of the internet at any current point. This recreated map would be a representation of how different internet entities are connected to each other, which can assist in showing the paths used by internet traffic as it flows from one point to the other. Internet maps come in different levels of granularity, i.e. whether one wants to view router level, Autonomous System (AS) level, or Point of Presence (POP) level maps. An Autonomous System is a single organization with their own network of routers. Examples include internet service providers and universities. A PoP, on the other hand, is a single geographical location/region where an AS has a set of network routers (some might have in more than one). There are several reasons why one would map the internet. Firstly, it is done to understand internet latency in different regions and find out why it occurs. Secondly, researchers have used such studies to uncover key insights concerning variances in internet latency levels across the different regions in the world, and why they are caused. Lastly, large content providing organizations like Facebook and Google, which need to deploy Content Distribution Networks to multiple locations, need to analyse the best CDN deployment strategies and their effectiveness.

The task of discovering an internet topology becomes more interesting when we focus on mapping the topology in Africa, which also happens to have the lowest internet penetration in the world [4]. Consequently, not much internet topology research has been done to accurately map out the internet landscape in the continent (save for a few notable works mentioned in the next section). Part of this is due to the fact that the common internet measurement platforms like Ripe Atlas, Speedchecker and CAIDA are heavily biased towards other western regions in terms of concentration of internet measuring probes [14]. Hence our task was to recreate the African internet topology using data from all 3 measurement platforms, and to represent it on a visualization platform that one can get information about the topology from.

3 BACKGROUND

3.1 The internet measuring and mapping process

Internet mapping is currently a multi-step process which involves making use of a couple of known hacks and tricks to come up with a close to accurate representation of the internet's topology. These include the use of common tools that make use of the *traceroute* utility (active techniques) and inferring from BGP tables (passive techniques) [1].

3.1.1 ***Active techniques.*** Using active techniques, the first step in mapping the internet topology is to conduct a series of internet measurements. Since the internet largely comprises of many proprietary networks that are independent from each other, it is almost impossible to simply get access to information about a network's topology; one would have to go past the enormous bureaucratical hurdles involved in trying to get topology information from all these autonomous system networks that belong to businesses and organizations. Hence, researchers have to use active inference tactics. The most common tool is the *traceroute* utility. *Traceroute* works by sending successive *ping* commands to a desired destination IP address from a source computer (called a probe) and collecting the ICMP time out messages returned by all the routers in the path from the source to the destination [8]. *Ping* is an internet tool that is used to test the reachability of a destination IP by sending 3 data packets to the destination and waiting for its reply. To prevent unending cyclic routing of packets in the internet, data packets have a time to live (TTL) variable, which is basically a counter that is set to a specific number when the packet is sent, and is decremented by 1 on each router it passes. When a packet with TTL=0 reaches a router, the router is configured to drop that packet and send back an ICMP time out response to the source, and this response often includes the IP address and hostname of the router. *Traceroute* then works by sending packets with successive TTL from TTL=0 up to the maximum needed to reach the destination. The first packet with TTL=0 reaches the first router in the path to the intended destination and that router returns an ICMP message together with its IP address. The next packet with TTL=1 does the same and terminates on the second router, and the process continues until the entire path of routers from source to destination is obtained [8]. A single probe (source from which *traceroute* will be run) would then send *traceroute* commands to multiple IP destination and combine all those path traces into one collection [1]. In a typical internet measuring project, one would make use of multiple probes to increase coverage [13].

Employing the above techniques yields raw measurement data that simply gives the IP paths from each measuring source to target destination. Further processing needs to be done to this data in order to make sense of it. Issues such as alias resolution (identifying sets of IP interfaces that belong to the same router) and anonymous router identification [6] need to be employed in order to map IP addresses to routers, and further heuristics then need to be done in order to group routers according to their respective Autonomous Systems (ASes) [2, 9, 11]. The end goal is to represent the internet topology as a graph data structure, with nodes representing individual routers (or ASes depending on the granularity desired) and links as the logical connections between them. Annotations then need to be added to the links and nodes. For the purpose of our study, we focused more on the Autonomous System (AS) map instead as it conveys more useful information about relationships between the different organizations and internet service providers at a continent and regional level.

3.1.2 ***Passive techniques.*** Passive techniques to infer the internet topology, on the other hand, work by consulting AS routing databases such as BGP Routing and RouteViews, which are regularly updated public databases that hold BGP announcements sent from different ASes [12]. ASes maintain peering relationships with other neighbouring ASes and that is what provides the functionality of the internet. These relationships are formed purely based on the individual policies and business interests of the ASes [10], and an AS would then regularly announce these relationships to a public database like BGP that keeps track of all such announcements [7]. If an AS wants to route traffic to another AS it would check BGP tables to see which AS routing path it can send the data to. Hence, as part of mapping the internet, one could consult such tables and infer the relationships between the ASes. This, however, only enables one to recreate an AS level map and not any other granularities. Normally, both passive and active techniques would be used in order to improve the coverage of the map discovered, since not both sources of information give the complete picture

4 RELATED WORK

Previous notable works that have embarked on a similar research have proven that mapping the African internet is a possible task, although not as easy. One of the first key research papers we looked into in great detail was Mapping the African Internet by Gilmore *et al* [5]. The research campaign carried out by the researchers shares multiple similarities with our own project: they were mapping the African internet at an AS and router level using both passive and active measurement techniques, and they also used Maxmind's Geolite databases to do their IP to ASN and City mappings. The only difference between our study and theirs was that they collected measurements from one vantage point in South Africa, and we sought to improve on that by collecting from multiple. Benefits of involving multiple vantage points were analysed in detail by Shavitt *et al* [13] and have also been proven by other African internet topology studies such as those carried out by Chavula *et al* [4] and Fanou *et al* [3]. The topologies that they produced were more comprehensive. However, they only made use from measurements from one platform, something which we also improved on by taking from three platforms for more comprehensive analyses.

5 DESIGN AND IMPLEMENTATION

5.1 Requirements gathering

Before we started development of the project, it was crucial that we get a solid understanding of all the user requirements we were to satisfy in our final product. The aim was to create an internet measurement and mapping visualization platform that can be used for both visualization and simulation. Visualization, in this context, refers to simply representing the gathered topology data structure on a map of the African continent, with nodes representing ASes (or AS Point of Presence) placed on their respective geographic locations on the map. The user would then be able to see various details about the topology such as the latency of each link and AS information about each node. It will need to be a static representation of the topology that reflects the latest set of internet measurements gathered; the measurement script will need to be running at set intervals in the background. The visualization will also need to allow one to select the measurement platform from which to get measurements in order to display on the map.

The simulation, on the other hand, would allow the user to start performing experiments and tests on the visualization. These tests could include manipulating the visualization by adding/removing nodes to simulate the emergence/closing of existing ASes, adding or removal of links to simulate link breakdowns and their impact on the traffic, and also to experiment with the impact of adding internet exchange points between ASes. However, the simulation is not the focus of this paper.

To gather these requirements, we had regular meetings with the supervisor (i.e. the client) via Jitsi where he briefed us of what our platform was to be able to achieve. Supplemental reading of past works also helped us understand the project and some of the processes to employ. This paper focuses on the process of transforming raw measurement topology data into a visualization that can map nodes and links between the different topology entities.

5.2 Software design process

To start the software design and development process, we decided on a Model View Controller architecture since our platform would constitute a user interface that is supposed to be easy and smooth, with a heavy reliance on background logical functions to display on the UI. The user interface would be our view, the logical data abstractions such as the graph data structures that are used to represent the map being the model. The scripts that process raw measurement data into the graph objects that will be stored in a database would be the controller classes.

The next step was to transform the user requirements captured from our meetings into a set of user stories, from which we drafted some use case diagrams that would guide the design. We came up with the following use cases (given in the use case diagram below) namely:

(1) View measurements from particular platform.
(2) View topology information: node info, link info, ASN info.
(3) View point of presence cities of a particular ASN.
(4) Zoom in to see POP level map.
(5) Zoom out to see city level map.
(6) View Internet Exchange Points

These use cases would become the basis for design and guide decisions such as which granularity of map to choose and which techniques to use to infer connections between map elements.

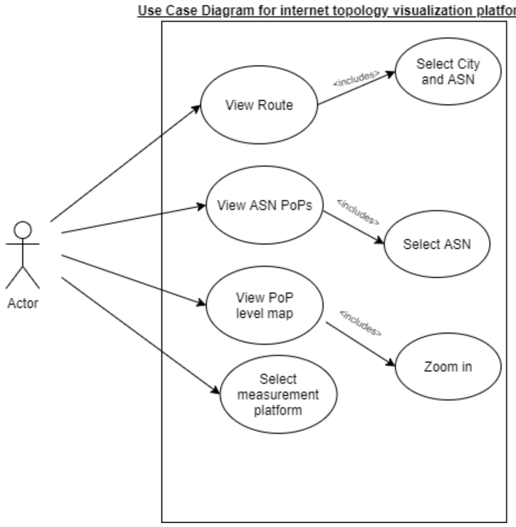


Figure 1: Use Case diagram showing the various use cases required

The last step in the design process was to design the sequence of transforming the raw data into a topology graph using a combination of some of the techniques learnt from the literature review phase.

5.3 Approach

Our project focuses on a city and PoP level AS map of the ASes across Africa, with functionality to zoom and view Point of Presence nodes for each Autonomous System in each city. This is because an AS is not really a geolocatable entity as it simply represents a network owned by an organization, hence will span over a wide area. However, ASes would usually have clusters of networks in different countries/cities, and this is referred to as a Point of Presence. We also decided to use the active measurement techniques to infer the topology connections. To achieve that, we decided to use the router level traces and iterate through them looking for points where hopping from one router to the next produces a change in the ASN (we would have done a router to ASN mapping first), then come up with an ASN graph. The process and all the tools used is described more detailed below.

Traceroute results from the measurement stage come as raw text format in JSON structure and need to be parsed for relevant information. Most of the JSON structure is irrelevant to the project. Measurements were coming from three platforms namely Speedchecker, Ripe Atlas, and CAIDA, and they all provide measurement results in different JSON structures. So, our scripts had to cater for all three.
The next sub-sections below outline the exact sequence taken to transform the raw data into nodes and links.

5.3.1 *Inserting data into MongoDB.*

We used Mongo DB Online for database storage of the traces. Our choice of database was because our data was in a document whose structure is heavily nested. To cater for three measurement platforms, we have one database with three separate collections. Each measurement result from a particular probe consists of a probe identifier (ID), and an array of results. Each result would be another document representing a single trace set. It would have among all its information a source IP address, destination IP address, and an array of traceroute hops. Each *traceroute* hop would have an IP belonging to a router in the path, sometimes a hostname, and Round Trip Time (RTT) values. The results (in JSON format) are streamed from the measurement API GET responses into the database collections

Before inserting the results into a collection, the script first performs an IP to ASN mapping for each IP in each array of traces. We used Maxmind's GeoLite2ASN database with a Python client to perform the queries. We would then append the ASN to the trace hop. The script does the same for the city geolocation of the IP, using Maxmind's GeoLite2City database. Where entries for the IP's do not exist in the database, we simply appended a blank ("") value for the field. So, it can be seen here how the extent of coverage for our discovered topology was limited to the reliability of the two databases from Maxmind. After appending those two values to each IP in the result, we then inserted the document in the Mongo database

5.3.2 *Removing null IP's.*

The next step in the process was to remove any IP addresses that were null from the *traceroute* hops. This is a means to clean the collections and make them easier to query. Unfortunately, since *traceroute* depends on the reachability of IP addresses, a good number of traces come with null IP's to indicate unreachability.

5.3.3 *Inferring connections from ASes from the traces.*

The next crucial step was to infer the connections between each AS represented in the traces. As per the gathered user requirements that the map needed to show city level nodes and PoP level nodes when zoomed in, we had two node datasets to create: the first having [ASN,City] combinations as nodes to help in graphing Point of Presence, and the second being City locations as nodes. After having mapped all IPs to ASN and City (those found in the Geolite database), the links can now be inferred. A function in the script iterates through each array of trace hops, and for each pair of successive IP hops it checks if there is a change of either ASN or City from one hop to the next. If there is, the two [ASN,City] combinations are identified as separate nodes and a link assumed between the two. However, if any of them lacks the ASN and/or City, it is simply skipped until the next valid IP hop is encountered. (The limitation of this method has been discussed in more detail later on in the Future Work section). The results of this process are added to the database as a separate collection.

5.3.4 ***Geolocating the nodes.*** After the stages described above, we had two MongoDB collections that can allow us to move to the next step towards having a visualization. The first one contains the source and target ASN and City together with the RTT value of that corresponding link. The second one contains the unique nodes gathered from the iteration as [ASN,City] combinations. The next step became to geolocate the nodes (and cities too).

However, geolocating an AS is not as trivial a task because an AS is an entire network of routers under one organization. Hence this could cover a large area, and could be in different cities too. So for the purpose of our map, the first naïve approach we tried was to pick a random IP address from each ASN and geolocate it. It worked to some extent, but one of its primary weaknesses was that two routers that are just next to each other but from different ASN's could be chosen and it would not reflect well on the map. The next approach was to try and take all the IP addresses under a particular [ASN,City], geolocate them all, then take the average longitude and latitude of all of them and use that as a kind of centroid of the AS Point of Presence. It proved to be better than the previous one; however, we noticed that for some ASN's that are in the same city, their IP addresses tend to have average geolocations that are so close that they would appear to be on top of each other on the final map. This is unwanted since it reduces the accuracy of our visualized topology. The third and final method, was to randomly select a geolocation point around the city's centre for each ASN. The random function picks a random radius and angle to place the point away from the city centre. That way we were able to have ASN Point of Presence nodes scattered within a city, showing the connections between them. Then when it came to viewing the map at a city level, we simply used the city's geolocation to represent the node, with all the connections of all the ASN's in that city branching out. The results of this node geolocation process were added to the database as well.

5.3.5 ***Creating the visualization.*** For the final part - the visualization - we used a technology stack comprising of static HTML, CSS, and Javascript for the front end. For the map and other mapping functions, we made use of the Google Maps Developers API. For the back end, we had Flask running in the background to help with rendering the web pages in Python. Our controller classes were written in Python as it is best for scripting. We used MongoDB Online for our data storage.

When the user loads the index page, it starts by displaying the African map with a city level ASN graph. This map simply shows a node per city if one or more ASNs are present in it. When the user zooms in enough, it changes to displaying a PoP level map in the different cities. This follows from the Schneiderman's design mantra of visualizations. At the city node zoom level, the user can click on any node and be presented with an information box that shows information such as the ASN(s) represented by the node together with the organization name etc. They can also click on a link to see its link delay as calculated from the RTT data.

By default the platform displays Speedchecker data. One can select from the drop down list and the platform can show them the latest collected data. There is also a Simulate button that takes one to the simulation page where they can alter certain aspects of the map to perform experiments as desired.

6 USABILITY TESTING PLAN AND PROCEDURE

After building our platform, we had to rigorously test it to ensure that it is user friendly and does the intended job. For testing, we decided to both perform our own tests to test edge cases, and also to invite participants to test the platform against a set of use cases. Each of the two types of tests have their own purpose.

6.1 Participant Recruiting

The usability test was meant to test the intuitiveness of the platform's design. Therefore it was imperative that we recruit a diverse set of participants with ranging academic backgrounds and experience with software platforms. We would then give them just enough background knowledge about the purpose of the platform for them to complete the objectives we give them to do. Participants were recruited by inviting friends and fellow classmates to agree to test our product. We managed to recruit a total of 7 participants with backgrounds in engineering, commerce, and humanities. Due to the ongoing global crisis and the meeting restrictions currently in place, we decided to conduct our tests via video calls. For those participants in engineering whom we presumed to have better experience with software use, we gave the source code to them to run on their machines and asked them to perform the use cases while timing them and taking note of the difficulty or ease they had in locating various functions of the platform. On the other hand, for those who could encounter issues installing and setting up the necessary environment needed to run the platform, we had to use Team Viewer instead.

6.2 The testing procedure

Before we gave the participants the set of objectives to complete in the testing exercise, we had to equip them with the needed background knowledge. We gave them a short document that gave a brief and simplified introduction to graphs and how networks in the internet are interconnected to form such graphs and how they determine traffic flow. It then moved into defining Autonomous Systems and the exercise of mapping the internet's topology. Finally, users were introduced to how the software fits into all that theory to enable one to see how traffic flows between ASes across Africa.

To perform the tests, we gave the users a list of use case objectives to accomplish and measured the time they took:

(1) Find out how many ASes there are in Cape Town for Speedchecker, and compare with the two other platforms
(2) Find out how many ASes there are in Luanda for Speedchecker, and compare with the two other platforms.
(3) Find out which measurement platform shows the most detail in terms density of the map
(4) Find out which measurement platforms show an AS in Zimbabwe, and which organization this AS belongs to
(5) Find out how many destinations traffic from Luanda goes to
(6) Discover a path that goes out of the continent
(7) Find out how many cities the ASN 37100 has point of presence in
(8) Which ASes is ASN 37100 linked with in Johannesburg

These use cases were chosen such that they cover the whole range of possible ways you could get information from the visualization. Each test session was scheduled at a set time as a 30 minute period. We spent the first ten minutes introducing the test participant to the platform and allowing them to read the document that gives them brief background knowledge they need to have to participate in the test. After signing the ethical consent form, we then let the user start the test as we timed them. For those running the code on their local machines, we asked them to share their screen via video call on Zoom so we could watch them perform the use cases. We gave them five more minutes to explore the platform and get familiar with it before embarking on the use case objectives. Finally, when they agreed to have been familiar with the platform enough to start, we then timed them from the start of the objectives to the end, also keeping track of the time taken to complete each objective. We also looked out for any particular trends in the participants' use of the software which would be a good indication to us of whether our design was intuitive. After each participant completed their test, we asked them to fill out a feedback form on which they ranked the use cases on their difficulty to achieve.

6.3 Test outcomes

From the results of the user tests, we noticed that participants did not take long to find their way around the platform to get the information they needed to accomplish the use cases. The average time to complete the task was 241 seconds. Figure 2 below shows a breakdown of the time taken by each participant to finish each use case task, with the total of all use cases indicating total time for the test by each participant. The fact that the participants took so little time despite them having differing academic backgrounds and levels of software exposure was a good indication that our design was simple and easy to use to get information from.

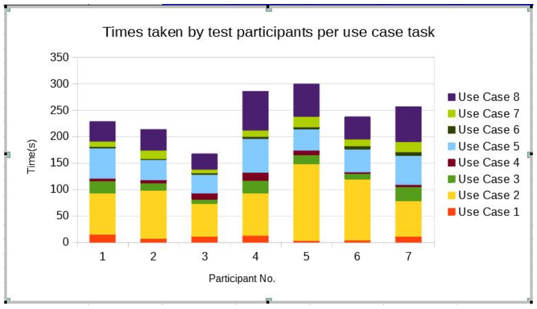


Figure 2: Participants testing times

On to the individual tasks themselves, we noticed that for task one most participants got the information easily by clicking on the Cape Town node whereas task two took longer as participants were not aware of where Luanda was located on the map. Most of them ended up finding later on that under View Routes mode one can select a city and have it highlight itself. However, that was an indication to us that we should make changes to our design to enable one to quickly locate a city without having to know where it is on the map. Also, we noticed that most participants took long on the last task of finding out which ASes a particular ASN is connected with in a particular city, which would have required the user to zoom into the city and physically count the individual ASes falling under the criteria. Some of them had forgotten that they could zoom in to see a POP level map. This was another indication to us that we should provide that hint on the interface to show the user how they can zoom into a city. Lastly, we noticed that users had some trouble accurately clicking on the city node icons as they appear small when zoomed out, and one of the feedback we got was that it would have been better if they were bigger or if they had a hover animation that enlarges the icon upon mouse over; those changes were considered too.

6.4 Limitations of the tests

The accuracy of the test results was compromised by a few factors. Firstly, for those conducting the tests via Team Viewer there was network latency involved from the use of Team Viewer. To that we also add the fact that most of those using Team Viewer were the non technical participants, whose entire user experience might be affected by that and thus affect the quality of feedback they give. Secondly, these are participants who might have had higher expectations of the performance of the visualization initially, as opposed to our more technical participants who understand software building processes and hence know how to prioritise functionality over aesthetics when testing a software. Lastly, we noticed that users had some trouble accurately clicking on the city node icons as they appear small when zoomed out, and one of the feedback we got was that it would have been better if they were bigger or if they had a hover animation that enlarges the icon upon mouse over; those changes were considered too.

6.5 Other tests involved

The other tests that we performed ourselves, on the other hand, were meant to test the edge cases of the software and ensure there are as few bugs as possible. To do this, we drafted a list of use cases that contain edge behaviours such as the following:

- Error recovery if new measurements data is not fetched on time or at all
- Inconsistencies in fetched data (null values etc.)
- Error recovery when network is interrupted

- Different combinations of event sequences that require the code design to be robust to not break. These include zooming in and out while in different modes, switching between different modes while zoomed in etc.

7 RESULTS OF THE VISUALIZATION PROCESS

This section reports on all the outcomes and results of the visualization process as a whole.

7.1 The map as a whole and the structure of the discovered topology

Our methodology managed to successfully uncover a decent representation of the African AS topology and enabled us to produce a graph representation on the map, with all the inferred connections and expected attributes shown. Figure 3 shows an example topology created from data from Speedchecker, showing the nodes as downward pointing arrowheads and the lines in blue as links. We used geolocation to place each respective AS Point of Presence in their respective cities and used lines to show which other ASes they are connected to. To avoid much clutter and aid visibility, we implemented the visualization such that when zoomed out, the map shows one node icon per city representing one or more Autonomous Systems which have a Point of Presence in that city. Hence, any links branching out of that node belong to one or more ASes. To view the ASes in a particular city, the user can either click on the node of interest or zoom in closer into the city until the map changes to show individual ASes as small circles connected to each other, each with a label that shows the AS and city it belongs to. The user can also switch between the three different measurement platforms and see their respective maps by selecting from the drop down list at the top left of the screen. (More images of the platform in use can be found in the Appendix section).

Some regions will show a high concentration of nodes (e.g Johannesburg/Pretoria region, Morocco region for the Speedchecker map). Johannesburg showed a high concentration of ASN's across all three maps from all measurement platforms, showing how heavily interconnected the region is.

Clicking on a node icon will display an information box to the left of the screen with the name of the city and also show more information about which ASes are present.

7.2 Getting relevant information from the visualization

We also built in sufficient implementation into the visualization platform for a user to extract useful information from the visualized data in line with the gathered user requirements. Much effort was put into trying to make the platform as helpful as possible.

The first major feature was the View Routes feature which allows a user to select from a list of sources a source city node and ASN, and the visualization draws the different paths from that AS point of presence to the different ASes it exchanges traffic with as measured from the traces. To implement this, we recorded for each trace result obtained from the measurements the specific AS path traffic took from the individual probes used in the measurements, and the probes became the sources in the visualization. To use this feature, a user would click the View Routes button, and the map responds by clearing all the links between all city nodes so as to enable the user to see individual paths. The user can then select from a drop down list on top of the map the city and ASN they would like to be their source. They can also click on the node and select the ASN to achieve the same effect. The map then draws the paths to the various destinations collected for that node, showing pop ups for the nodes it passes through. The user can repeat this process for as many times for different nodes. Once satisfied, the user can click the Exit View Routes button to restore the original visualization. Figure 4 shows the feature in use, with the user having selected Cairo as the source for ASN 8452, then it shows the trace paths to the different destinations. It is also worth noting that while in this mode, the View PoP button gets greyed out and disabled.

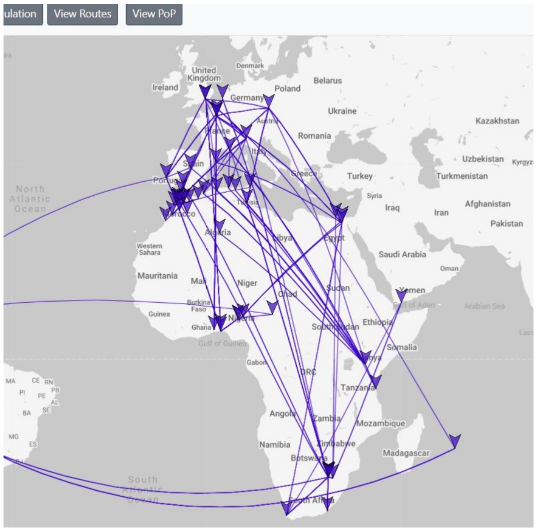
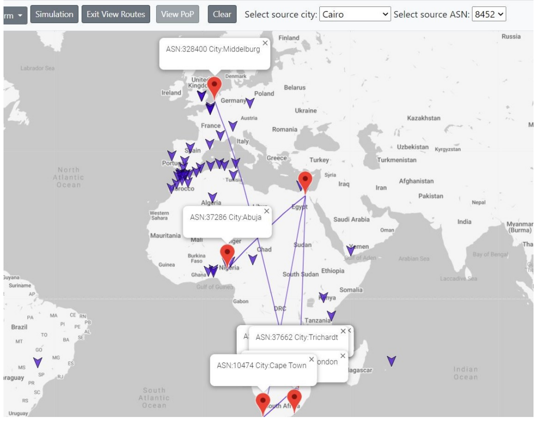


Figure 3: Discovered topology for Speedchecker



Select source city: Cairo Select source ASN: 8452

Figure 4: View Routes mode viewing paths from Cairo to Nigeria, Cape Town, Johannesburg

The second major feature was enabling the user to select an ASN and have the visualization show which cities it has point of presence in. This will be shown by the respective city nodes changing their icons temporarily until a new ASN is selected or the mode is exited. Figure 5 below shows point of presence nodes shown for ASN 37100. The Clear button would remove all red nodes and pop up windows to allow the user to start afresh. This feature will be useful for researchers who would like to view the coverage of specific Autonomous Systems. Another similar feature is allowing the user to see which ASN's have presence in a particular city; this they can do by clicking on the respective city node and the information gets displayed in an information window on the screen.

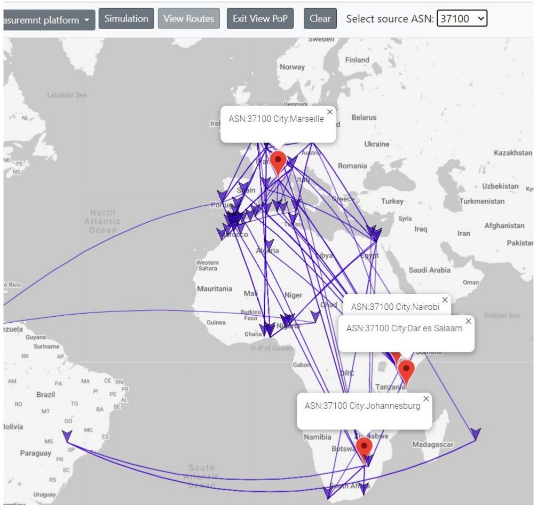


Figure 5: View PoP mode showing point of presence cities for ASN 37100

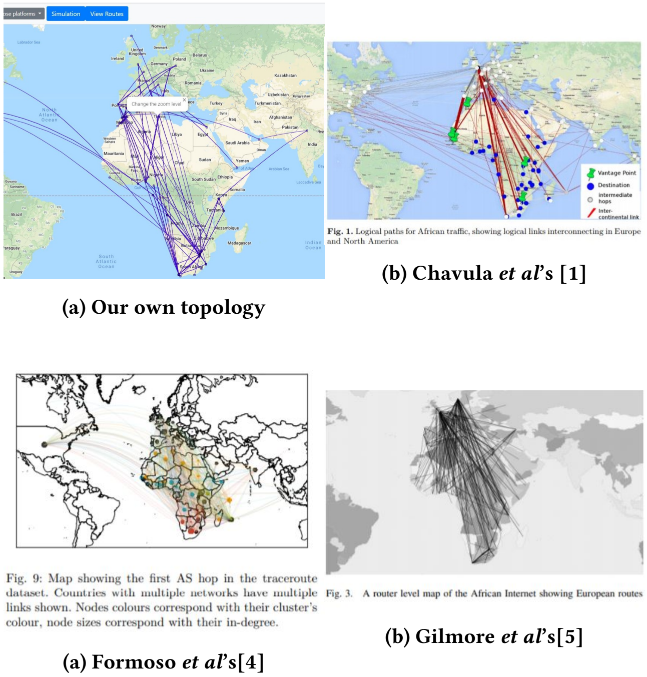
Lastly, there is a Simulation button that would redirect the user to the simulation part of the platform where the user can perform experiments and simulations on the internet topology. Specifics about the simulation were covered in another separate paper and were not the area of focus for this particular paper.

8 DISCUSSIONS

We use this section to analyse and discuss the effectiveness of our visualization process and outcome in terms of accurately depicting the state of the African internet and providing sufficient information to a user of the platform.

8.1 Comparing with previous works

The first thing that gave us assurance that our results were in the right direction was the circuitous nature of some internet traffic routes originating from the continent, as discovered by the past researchers we reviewed. Chavula *et al*[1], Formoso *et al*[4] and Gilmore *et al*[5] in their papers did a similar study of mapping the African internet and found out the same result concerning outbound traffic. They discovered that most traffic originating from Africa and destined to other African locations tends to be routed to other continents first via circuitous routes. Chavula *et al*[1] attributed this phenomenon to lack of sufficient inter-IXP peering among ASes in the continent. As a consequence, users in Africa experience higher latencies and lower quality of internet experience than others. Better service of internet can be provided if there is sufficient peering among ISP's. The structure of our visualized topology to a great extent reveals this characteristic, as shown by some of the links originating from Southern Africa and going to countries like France, the UK, Italy, some states in the USA, and Portugal. Using the View Routes mode on the Speedchecker map, we even notice a route from Mauritius that goes first to Paraguay before returning to Johannesburg.


Simulation
View Routes
Vantage Point
Destination
intermediate hops
Inter-continental link
Fig. 1. Logical paths for African traffic, showing logical links interconnecting in Europe and North America
Fig. 9: Map showing the first AS hop in the traceroute dataset. Countries with multiple networks have multiple links shown. Nodes colours correspond with their cluster's colour, node sizes correspond with their in-degree.
Fig. 3. A router level map of the African Internet showing European routes

(a) Our own topology

(b) Chavula *et al*'s [1]

(a) Formoso *et al*'s[4]

(b) Gilmore *et al*'s[5]

Figure 7: Comparisons of our topology with three other past researchers.

Figure 7 gives a side by side comparison of our map (taken while the platform was still under development with an earlier set of data) and three other maps. Even though Gilmore *et al*'s[5] was a router level map, it still reveals a roughly similar structure as would an AS level map would. Formoso *et al*'s[4] main focus is to show the various concentrations of nodes per country, and after comparing with our map it was observed that our maps agree in reflecting high node degrees in Southern Africa and North Africa into Europe, as well as slightly less node degrees in the West African countries of Ghana and Nigeria.

Secondly, comparing our topology with these three maps, we can see that ours follows the same structure as them in terms of how the connections are oriented and where they are concentrated. To be more specific, our map shows how most traffic connects in the Johannesburg region and this is true for the maps from all the other measurement platforms. The map also shows a considerably strong connection presence in Cape Town as well and much connection between the two cities. Then from there, a considerable amount of traffic then crosses the continent into Europe just above Africa, before coming back into the continent, which agrees with the maps that [1] and [5] discovered. Our map also shows less stronger connection presence in countries such as Nigeria, Ghana, Kenya, Angola, and Morocco just to mention a few. These countries, unlike South Africa, have much less Point of Presence nodes per country, which can be seen in the map produced by [1] which shows the strongest node degrees being in South Africa. Lastly, we also discovered links to India and the United States as did [4] and [1].

We also compared our topology with Fanou *et al*[3] who did a detailed study on the African web ecosystem, and in the process produced a map that geolocated Google cache servers across the continent. Although it is a different map from what we visualized in terms of focus, the two maps when compared show a similar trend. The presence of Google caches in the countries indicated corresponds to the presence of AS connections in the regions discovered in our visualization, with concentration mostly in Southern Africa, Morocco, the Europe region, and Nigeria just to mention a few. All these observations go on to show that our map visualization process was to a great extent effective given the limitations we had in doing the project (explained more in the Conclusions section).

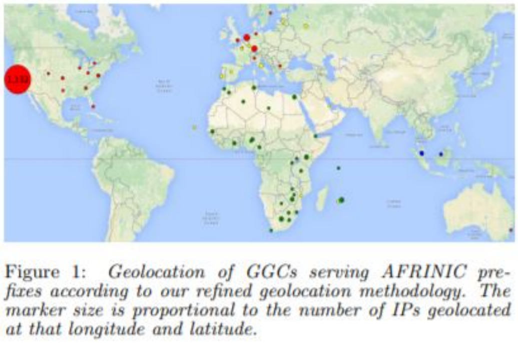


Figure 1: *Geolocation of GGCs serving AFRINIC prefixes according to our refined geolocation methodology. The marker size is proportional to the number of IPs geolocated at that longitude and latitude.*

Figure 8: Location of Google Caches by concentration in Africa [3]

However, our visualization did not depict as much outbound traffic as did the other papers we are comparing with. This is because our focus was on connections within the African internet more and hence our choice of probes and destination IP addresses was from Africa, causing most of our discovered traffic to be within the continent. Our map also failed to discover links with Madagascar, Seychelles, and other islands to the east of Madagascar which are part of Africa and should have been discovered. This is possibly due to some of the limitations to the accuracy of our work that are explained further in the Conclusions section.

8.2 Comparing the 3 measurement platforms in terms of topology coverage

The next step of our analysis is to compare the three measurement platforms in terms of how much of the topology they discovered. Our first step in doing this was to simply compare the final topologies from all three platforms and see which one gave the most coverage. Looking at them side by side as in Figure 9 below, we can see that Speedchecker has the most number of nodes and links, followed by RIPE and CAIDA having almost similar coverages. The differences are mostly due to the platforms having different biases towards certain regions in terms of the number of probes they have there [14].

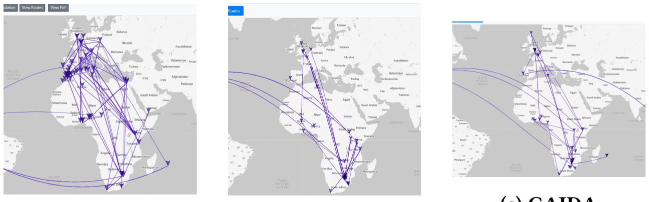

(a) Speedchecker **(b) Ripe Atlas** **(c) CAIDA**

Figure 9: Comparison of topologies from the 3 platforms

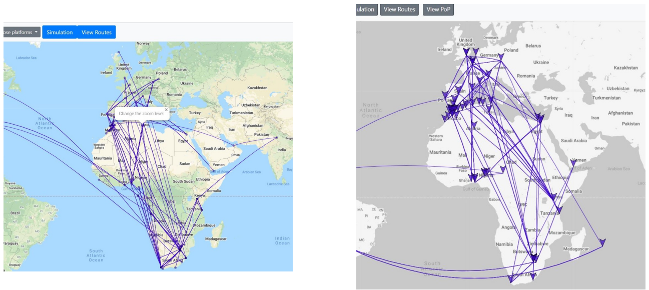
8.3 Discussing on the effectiveness of the visualization produced using different metrics

In this sub-section, we discuss on how effective the visualization we produced was in meeting the design requirements an fulfilling its intended purpose of aiding internet researchers to gain insight about the African internet topology.

The initial aim of this project was to create a platform that shows on a geolocated map the present state of the African internet topology at any time, taking regular internet measurements data from 3 platforms and updating the map as required based on the new data. This platform would not only show the topology state of the map, but also allow researchers to draw as much useful information as they can about the topology from interacting with it. Hence, features would need to be built into it that allow for that.

8.3.1 ***Effectiveness of having a regularly updating map.*** The first novel design aspect of the platform was to ensure a regular update of the AS level map every 4 hours when it is being run. It achieves this by running a script that makes API calls to the three measurement platforms to perform internet measurements every 3 hours. When the results get fetched, they are reflected on the map upon refresh. This would be very effective for anyone who wishes to do studies of the internet topology in the continent, as they can just see the current state of the topology at any point. It would cut out the need for anyone to perform their own internet measurements for them to get information about the topology or perform simulations. Potential uses which could benefit from this self-updating visualization mostly include studies of the internet that do not directly involve conducting measurements such as analysing content distribution network coverage and deployment [14]. As further improvements get added to the platform, it could become an indispensable tool for people who wish to study anything that relates to the internet's topology.

We also performed studies to see if the self updating capability of the platform actually discovers new paths and links and reflects changes over different iterations. The results we got were encouraging, and are demonstrated in Figure 10. We compared two



(a) Earlier dataset

(b) Later dataset

Figure 10: Speedchecker topologies from 2 successive measurements datasets

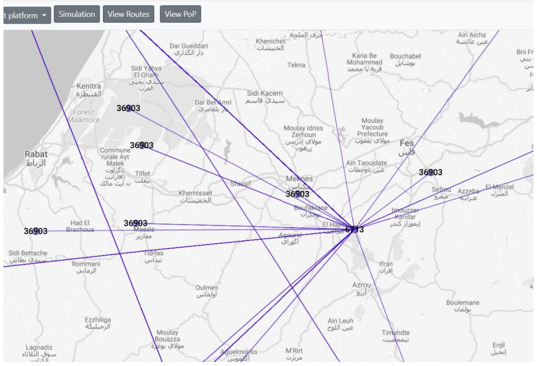


Figure 11: POPs in Morocco with good inter peering

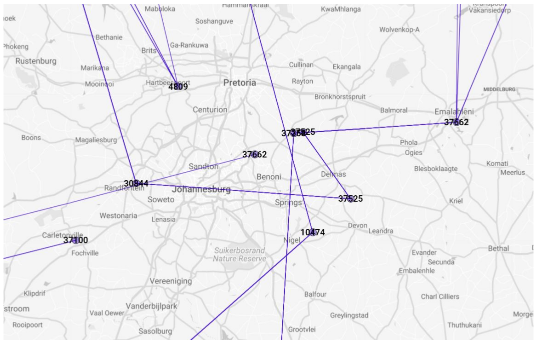


Figure 12: POPs in Johannesburg with little inter peering

Speedchecker topologies from successive iterations and we noticed that the updated map discovered additional links and nodes that the previous had not. For instance, we noticed that the second visualization process discovered circuitous links from Mauritius to Johannesburg and Nigeria via Sao Paulo in Brazil and Colombia respectively. The latter visualization also discovered more links from Kenya and Egypt than before. However, the new topology seemed to have lost some of the Johannesburg and Cape Town links. A better future implementation of the visualization would be one that simply updates the map with additional links and keeps the previous ones instead of discarding them.

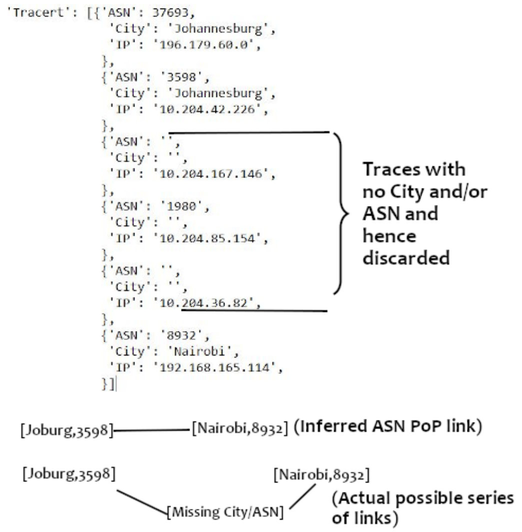
8.3.2 ***Effectiveness of having a City level and POP level maps in giving useful information.*** Secondly, we turn to analysing how effective our implementations of a city level map and POP level map were in fully communicating the state of the internet topology in a meaningful way. Visualizing the city level map was a less challenging process as all that was needed to be done was to group all ASN's that belong to a particular city into one node and group their outgoing links as well. As discussed in the previous section, the resulting city level map was seen to be accurate in terms of the structure of the topology when compared with other notable research papers. The POP level map, however, was a little different. As mentioned in the Implementation section, geolocating an AS is not a trivial task and so is representing linking relationships between two ASes properly. This is because an AS is a logical entity and less of a physically locatable entity. However, for the purpose of this visualization, we ended up picking random locations around a city centre and place each AS that has a presence in that city on its own location, and represent it as a small circle with the ASN as a label on the map. The other limitation currently facing our POP level map with regards to giving as much information as it should is the fact that within the cities themselves there is very little inter-AS peering. As a result, most nodes when zoomed in just show connections going out of the city and very little to none among the POP's of each city. Hence our POP level map is restricted to merely showing one the number of AS POP's each city has and not the interconnections within the city because they barely exist, save for a few cities like Morocco where they do.

On a brighter note, the interactive design of the visualization has some very helpful features that would aid researchers in deriving information about the internet topology in Africa. There is the View Routes feature which, as mentioned in the Design and Implementation section, allows one to select a source node (ASN and City) and have the map draw out the paths taken by traffic from that source to the different destinations collected in the traces. The map will give the user rich details about the path such as the cities and ASN's it passes through as well as the recorded Round Trip Time (RTT) for the path. This would be very useful in giving information about the peering of ASN's across POP's. We decided to add it since we realized that showing a user a map alone with connections will not convey full information about the topology in terms of where traffic originates and destines towards when making those connections shown by the map. It also serves to give insight into which nodes mostly receive traffic and which ones mostly generate or transit. Added to that is the ability to click on individual icons and get a list of ASN's and their organization names they represent on that location. The visualization also has a feature that allows a researcher to select an ASN and visually see which cities it has point of presence in, which would be useful in seeing which ASes have the most penetration in certain regions for instance. Further studies could then even be made about such ASes to find out if they are local or overseas companies. From there one could start using that to explain the circuitous nature of some of the routes for instance.

9 FUTURE WORK

For future improvement it would be more ideal to use more reliable databases to do ASN/City to IP mappings, since our nodes in the map were ASN, City combinations for every IP. Examples of commonly used ones include the WHOIS database and BGP Stream which shows AS peering information. Another future improvement would be to include other ways of visualizing AS relationships that are not only geographical, since the AS topology is more of a logical topology than a physical one. An example of such would be a 3D hyperbolic visualization as used by [5] in their paper.

The final major limitation of our study was the way in which we inferred connections between ASes using our algorithm. The algorithm iterates each traceroute hop looking for changes in ASN and City for each hop and recording those as separate nodes and links. However, since hops with no ASN nor City are discarded, it is highly possible that some links in between nodes could have been falsely inferred as shown in the pictorial representation below. Hence, making use of more reliable databases and incorporating passive measurement techniques would improve the accuracy of the map.The figure below illustrates that best with an example traces array showing trace hops between different IPs. The Maxmind Geolite databases are then used to map the IPs to ASNs and Cities and if either one or all are missing, the hop is disregarded and a link is directly assumed when the next valid hop is reached.



```
'Tracert': [{'ASN': 37693,
             'City': 'Johannesburg',
             'IP': '196.179.60.0',
            },
            {'ASN': '3598',
             'City': 'Johannesburg',
             'IP': '10.204.42.226',
            },
            {'ASN': '',
             'City': '',
             'IP': '10.204.167.146',
            },
            {'ASN': '1980',
             'City': '',
             'IP': '10.204.85.154',
            },
            {'ASN': '',
             'City': '',
             'IP': '10.204.36.82',
            },
            {'ASN': '8932',
             'City': 'Nairobi',
             'IP': '192.168.165.114',
            }]
```

Figure 13: Illustration of how false link can be assumed due to missing data

10 CONCLUSIONS

In this project, we sought to recreate the African internet topology at an Autonomous System level and represent it on a map using internet measurements data collected on a regular interval daily by the back end implementation of the platform. The goal was to see if, firstly, it was possible to do so, and secondly, how accurately we would be able to recreate the topology. The visualization platform needed to also allow the user to select from a three measurement platforms (Speedchecker, RIPE, CAIDA) as for the choice of measurement data to visualize. To judge the effectiveness of our topology mapping, we were to compare our discovered topology with those done in previous works we referenced in our literature review. Furthermore, we were to conduct various tests that analyse whether the visualization can be used to extract useful information from a research perspective.

Overall, our internet mapping process was successful in creating the intended topology. Using the algorithm outlined in the Approach section, we managed to infer most of the relevant AS connections, subject to the reliability of the used traceroute probes and databases consulted. The user interface we created enables a user to extract relevant information about the structure of the internet topology in Africa with ease. For instance, one can easily see where most internet nodes are concentrated and also the structure of the links between AS'es. Our map shows the circuitous routes that heavily characterize the African internet topology due to lack of sufficient ISP peering, which is something that was expected from the literature review we did. This is another sign that our mapping process was accurate.

REFERENCES

[1] Chavula, J., Feamster, N., Bagula, A., and Suleman, H. Quantifying the effects of circuitous routes on the latency of intra-africa internet traffic: A study of research and education networks. 64–73.

[2] Claffy, K., Hyun, Y., Keys, K., Fomenkov, M., and Krioukov, D. Internet mapping: From art to science. In *2009 Cybersecurity Applications Technology Conference for Homeland Security* (2009), pp. 205–211.

[3] Fanou, R., Tyson, G., Francois, P., and Sathiaseelan, A. Pushing the frontier: Exploring the african web ecosystem. In *Proceedings of the 25th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2016), WWW '16, International World Wide Web Conferences Steering Committee, p. 435–445.

[4] Formoso, A., Chavula, J., Phokeer, A., Sathiaseelan, A., and Tyson, G. Deep Diving into Africa's Inter-Country Latencies. (English).

[5] Gilmore, J., Huysamen, N., Cronje, P., de Klerk, M., and Krzesinski, A. Mapping the African Internet. (English) [On recreating and mapping the african network].

[6] Gunes, M. H., and Sarac, K. Resolving anonymous routers in internet topology measurement studies. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications* (2008), pp. 1076–1084.

[7] Gupta, A., Calder, M., Feamster, N., Chetty, M., Calandro, E., and Katz-Bassett, E. Peering at the internet's frontier: A first look at isp interconnectivity in africa.

[8] Huffaker, B., Plummer, D., Moore, D., and Claffy, K. Topology discovery by active probing. In *Proceedings 2002 Symposium on Applications and the Internet (SAINT) Workshops* (2002), pp. 90–96.

[9] Hyunseok, C., Sugih, J., and Walter, W. Inferring as-level internet topology from router-level path traces. *Scalability and Traffic Control in IP Networks* (2001).

[10] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking 9*, 6 (2001), 733–745.

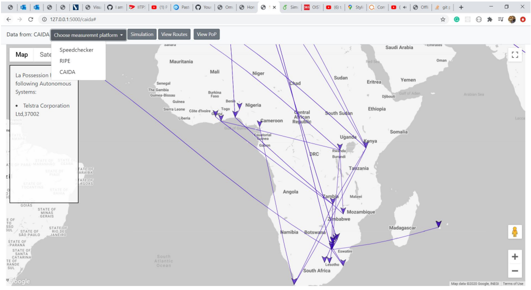
[11] Magoni, D., and Hoerdt, M. Internet core topology mapping and analysis. *Computer Communications 28*, 5 (2005), 494–506.

[12] Nicholes, M., and Mukherjee, B. A survey of security techniques for the border gateway protocol (bgp). *IEEE Communications Surveys Tutorials 11*, 1 (2009), 52–65.

[13] Shavitt, Y., and Weinsberg, U. Quantifying the importance of vantage point distribution in internet topology mapping (extended version). *IEEE Journal on Selected Areas in Communications 29*, 9 (2011), 1837–1847.

[14] Singh, R., Dunna, A., and Gill, P. Characterizing the deployment and performance of multi-cdns. In *Proceedings of the Internet Measurement Conference 2018* (New York, NY, USA, 2018), IMC '18, Association for Computing Machinery, p. 168–174.

A SCREENSHOTS OF USER INTERACTION WITH THE PLATFORM DURING TESTING

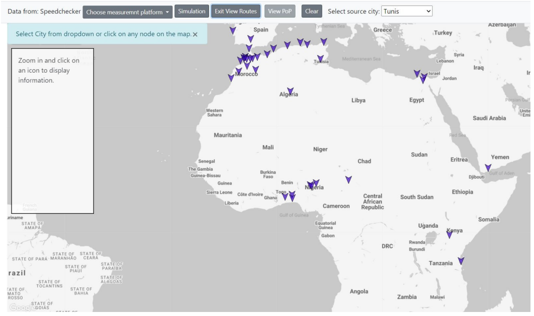


Data from: CAIDA

La Possession following Autonomous Systems:

- Telstra Corporation Ltd,37002

Figure 17: User about to select new measurement platform



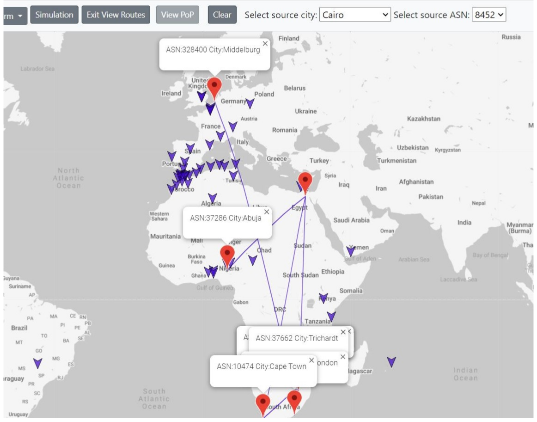
Data from: Speedchecker

Select source city: Tunis

Select City from dropdown or click on any node on the map. ×

Zoom in and click on an icon to display information.

Figure 14: View Routes mode before user selected City and ASN



Select source city: Cairo

Select source ASN: 8452

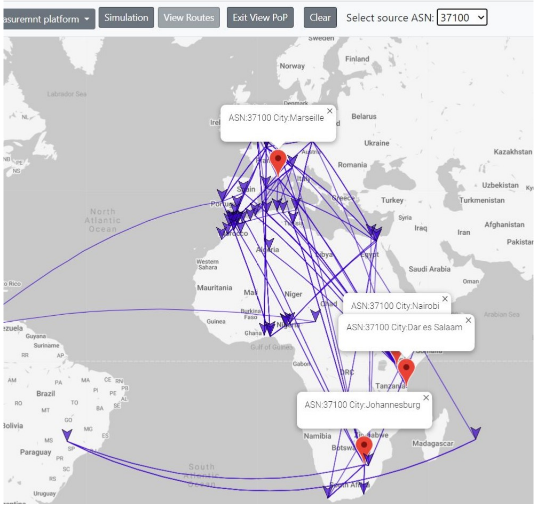
ASN:328400 City:Middelburg

ASN:37286 City:Abuja

ASN:37662 City:Trichardt

ASN:10474 City:Cape Town

Figure 15: View Routes mode after user selected City and ASN



Select source ASN: 37100

ASN:37100 City:Marseille

ASN:37100 City:Nairobi

ASN:37100 City:Dar es Salaam

ASN:37100 City:Johannesburg

Figure 16: User viewing PoPs for an ASN of choice