CS/IT Honours Final Paper 2023

Title: Mapping and Visualizing the Router-Level Internet Topology

Name: Maganyase Elias Sihlangu

Project Abbreviation: MAIT

Supervisor: Dr Josiah Chavula

Category	Min	Max	Chosen
Requirement Analysis and Design	**0**	**20**	**20**
Theoretical Analysis	**0**	**25**	**0**
Experiment Design and Execution	**0**	**20**	**0**
System Development and Implementation	**0**	**15**	**20**
Results, Findings and Conclusion	**0**	**20**	**10**
Aim Formulation and Background Work	**0**	**15**	**10**
Quality of Paper Writing and Presentation	**10**	**10**	**10**
Adherence to Project Proposal and Quality of Deliverables	**10**	**10**	**10**
Total marks		**80**	**80**

Source Code:
https://drive.google.com/drive/folders/11fkWW_yHTXEadekbqIWOKOlaFoiZLBH_?usp=sharing

Mapping and Visualizing the Router-Level Internet Topology

Maganyase Elias Sihlangu

Shlmag002@myuct.ac.za

ABSTRACT

Our research employed a multi-faceted approach to address the challenges associated with internet mapping at the router level. Among others, these include: the aggregation of routers within the same address range, representing vast data points within confined visual spaces, optimising the responsiveness of the visualisation in the user interface, and filtering private IP addresses to maintain focus on the public internet topology. Tools, including RIPE ATLAS, graph databases and various geolocation databases, were leveraged in the development of the topology visualisation tool.

Additionally, our methodology emphasized user-centric design, ensuring that the resultant visualisation remains usable and accurately reflect the structure of the router level topology. We incorporated advanced features, such as regional filtering and zoom capabilities, to enhance the usability of the tool.

The outcome is a periodically updating visual tool that maps Africa's router-level internet topology. Furthermore, our study opens avenues for future research in optimizing visual representations, mapping network topologies.

CCS Concepts

- **Human Computer Interaction → Visualisation**
- **Networks → Network representation; Topology mapping**

Keywords Network Topology Discovery; Traceroute; Topology Visualisations; Graphs; Autonomous Systems; Geolocation; Internet Routes; User-Centred Design

1. INTRODUCTION:

The Internet is fundamentally an interconnected network of networks, organized through various layers of topologies and held together by protocols [1]. These are sets of rules that govern the exchange of data and communication between networks [1]. Critical to the understanding the structure and functioning of the Internet is the Router-level topology, which is a representation of the interconnected network of routers that make up the global Internet infrastructure [1]-[3]. This topology illustrates how routers are connected to each other to facilitate the flow of data packets across the Internet [3].

At a higher aggregation level, the Internet comprises numerous Autonomous Systems (ASes). An AS is essentially a collection of IP networks and routers under the control of a single organization that presents a common routing policy to the Internet [3]. Given that the Internet is a decentralized entity with thousands of connected networks, ASes serve as the building blocks that ensure structured and regulated communication [3]-[5]. Each AS is assigned a unique AS number (ASN), ensuring differentiated and structured communication across the decentralized Internet structure [3].

The essential function of guiding data packets through the paths of the Internet is defined as routing. Routing, in the context of the Internet, pertains to the process by which data packets are directed from their source to their destination across potentially multiple intermediary points (routers). One of the pivotal protocols that underpin Internet routing is the Border Gateway Protocol (BGP) [1][2][4]. BGP is the standardized exterior gateway protocol designed to exchange routing and reachability information among ASes on the modern Internet. BGP's primary function is to establish and maintain pathways (routes) between the various ASes, ensuring data packets can traverse the breadth of the Internet, often crossing multiple AS boundaries [1][4]. BGP presents itself in two primary forms: eBGP (Exterior BGP) and iBGP (Interior BGP). eBGP facilitates the exchange of routing information between distinct Autonomous Systems (ASes), enabling them to disseminate details about the IP addresses they manage and the best routes to access these addresses. Conversely, iBGP operates within an individual AS, where multiple routers may exist [1][4]. Its role is to circulate routing data internally, guaranteeing consistency and efficient data packet movement. BGP's significance in the underpinnings of the Internet is paramount, acting as the cohesive force that integrates various ASes [1][4]. This ensures that the Internet's decentralized structure remains unified, preventing potential fragmentation or isolated communication channels [1].

1.1 Project Aims

While both AS-level topology and router-level topology mapping play a role in understanding the Internet's structure, they offer different perspectives. AS-level topology primarily focuses on how Autonomous Systems interconnect and exchange data on a macro scale, providing a high-level view of the Internet's architecture [1]-[4]. On the other hand, router-level topology pertains to the network of routes data packets traverse within and between these ASes [1]-[4]. This is a lower-level view of the internet. This paper specifically concentrates on the latter.

We are interested in answering the question "How can we accurately visualise the African internet topology at router level such that users are able to identify different routers making up the intra-continental topology, the geographic location of the routers, links between routers, logical paths between regions and countries and ISP/organisations that manage the routers. Furthermore, how can we improve the usability, accuracy, and reliability of the visualisation tool by leveraging user-centred design techniques?"

In this paper, we present the design and implementation of a tool that map and visualize Africa's router-level internet topology. Using the Ripe Atlas platform, we created an extensive dataset of traceroute measurements as the primary technique for topology discovery. The gathered data underwent processing so that it better suited for visualisation.

Furthermore, we conducted rigorous evaluations to validate the accuracy and efficacy of our tool. Expert opinions from faculty members conducting research in computer networks were sought to help validate the accuracy and soundness of the mapping technique employed. Moreover, usability tests were conducted, targeting a wider audience to assess the tool's usability and effectiveness in visualising the topology.

2. BACKGROUND

In the introduction, we touched on topology mapping specifically at the autonomous system (AS) and router levels. This focus is deliberate: global and continental internet topology mapping predominantly occurs at these two levels. This level of abstraction is driven by the insights stakeholders seek from continental-scale topology maps. At this scale, the primary interest lies in understanding inter-network communication: how networks exchange traffic and the routes packets take from their source to their destination networks [2]. The AS level effectively illustrates network-to-network communication, while the router level offers a detailed view of packet paths [1]-[4]. However, there exist lower and intermediary levels in computer networks. It is worth noting that mapping techniques at these levels often inform the methodologies used for AS and router level mapping [4]. In this section, we'll delve deeper into these abstraction levels and their significance.

2.1 Interface Level Topology:

This level of abstraction of the internet topology is concerned with network layer connectivity. Here, distinct IP interfaces of both routers and end-hosts are represented as nodes. a link between two nodes represents a direct network layer connectivity between them. It is important to note that at this abstraction level, devices operating at lower OSI layers, that is, layers lower than the network or IP layer (e.g., hubs and switches) are ignored [4].

Routers are typically designed to have multiple interfaces. As a result, a single router can appear multiple times in the topology representation, each instance corresponding to a different interface. Conversely, end-hosts are typically equipped with a singular interface, therefore they are represented as individual nodes in the topology map. However, interface level topologies often exclude end-hosts, focusing primarily on router interfaces. This helps to simplify the topology [4].

To discover the comprehensive topology at this level, data from individual traceroute measurements are aggregated. Each measurement outlines an IP path [4]. For a holistic view, results from various measurements are combined. This approach heavily relies on hosts capable of executing traceroute (commonly termed vantage points) and a list of target IP addresses. As part of this probing procedure, multiple vantage points initiate traceroute probes targeting a specified set of IP addresses. The outcome is an integrated interface-level topology, derived from the union of all observed IP paths. When depicted as a graph, the nodes in the graph present individual interfaces and links represent directly linked interfaces [3][4].

2.2 Router Level:

At the router level topology, the focus shifts from individual interfaces to the routers themselves and the devices that connect to the Internet. This topological perspective is derived by grouping together interfaces that are part of the same router. In this depiction, a node signifies an IP-compliant device, which can be a host, such as a computer, or a more complex device like a router that possesses multiple interfaces [1][3][4]. When visualizing connections, an edge (or link) between two nodes indicates that their respective devices share a mutual IP broadcast domain [4]. In essence, the Router level topology is the aggregate view of the interface level topology [4].

Similar to topology discovery at the interface level, a key method to discover the router level topology is through traceroute measurements [1]-[4]. Each measurements returns hops which correspond with the sequence of routers a packet visits en route to its destination [1]-[5]. ICMP packets are typically used as probes in these measurements [1][5]. However, there exist variants of traceroutes that utilizes UDP or TCP packets. Similarly, the measurements are initiated from multiple vantage points and directed towards predetermined targets [1]-[4].

The distinct IP addresses associated with each interface of the same router are referred to as aliases [3][4][6]. To derive a router-centric topology from traceroute data, alias resolution is employed. This process identifies different interfaces that belong to the same router and represent them as one node [3][4][6]. Given that multiple methods exist for alias resolution, we discuss them below.

2.3 Alias Resolution Techniques:

1. Common Source Address: In the process of determining if an IP address A has an alias, this method sends a probe to an unused port, expecting an "ICMP port unreachable" reply. The reply typically has the IP address of the router's shortest-path interface as its source address. If the source IP address of the reply message is different from A, these two IPs are aliases of the same router [3][4][6].

2.Common IP-Identification Counter: This method utilizes the packet ID in the IP header, primarily intended for packet reassembly after fragmentation. The core assumption is that a router has a unique IP ID counter. Thus, sequential packets from the router will have consecutive IP IDs, regardless of the departing interface. The Ally tool, featured in [4] and used in the Rocketfuel project, leverages this mechanism. Ally sends a UDP probe with a high port number to two potential alias IPs. The resulting ICMP "Port Unreachable" replies carry distinct IP IDs (x and y). A third packet sent to the initial responder's address, and assuming its ID is z, if the IDs follow the pattern $x < y < z$ with a small difference between z and x, they're deemed aliases [4][6].

3. RadarGun: It enhances the limitations of Ally by gauging the changes in the IP ID counter. Instead of separately testing IP pairs, it probes a list multiple time. IPs are inferred as aliases if their ID counters' "velocity" aligns. However, challenges arise due to potential counter wraparounds [4].

4. DNS-Name: Aliases can be inferred by examining DNS name similarities of router interfaces. It's effective if a consistent naming convention exists but struggles at AS borders where different conventions prevail [4][6].

5. Graph-Based Resolution Heuristics: By overlaying individual traceroute measurements on a graph, one can use heuristics like the "common successor" or "same traceroute" to deduce potential aliases [4][6].

6. Analytical Alias Resolution Using tools like AAR (Analytical Alias Resolver), one can analyse traceroute-derived paths. It examines common IP address assignment schemes to deduce aliases, especially effective for symmetric routes [4].

7. Record Route Option: employs conventional traceroute measurements but with the Record Route (RR) IP feature activated to identify IP aliases. During the initial nine hops, it records two interfaces: one from the outgoing route and another from the return route. Despite its intuitive nature, it's challenging to implement due to varied RR implementations by routers and the intricacy of merging RR and traceroute data [4][6].

2.4 PoP level topology:

The PoP (Point of Presence) level topology can be viewed as an extension of the router level topology. This extended approach focuses on aggregating routers managed by the same Autonomous System (AS) and situated within the same city or suburb [3][4][6]. By doing so, the PoP level topology offers a more consolidated view of the network.

The PoP level abstraction eliminates the need for alias resolution. Alias resolution primarily addresses the challenge of identifying multiple IP addresses that belong to the same router [4][6]. At the PoP (Point of Presence) level, the focus shifts from individual routers to a higher-level aggregation. In PoP level topology, multiple routers and devices within the same geographical area (e.g., city or suburb) and owned by the same Autonomous System (AS) are aggregated into a single node or point of presence. Since the PoP level already abstracts away from individual device identities, the granularity at which alias resolution operates becomes redundant [4][6]. Instead, the emphasis at the PoP level is on understanding the interconnections between these aggregated nodes rather than the individual router interfaces.

2.5 AS level topology:

At the AS (Autonomous System) level, the topology primarily consists of distinct networks [1]-[4]. There exist relationships between these networks. These include peering links, which are connections between two networks for direct traffic between them. There are also customer-to-provider relationships, where a customer network purchases access to the internet from an internet service provider. In some cases, there might be provider-to-provider relationships, indicating mutual agreements between providers. Additionally, Internet Exchange Points (IXPs) play a pivotal role at this level, serving as physical infrastructure through which multiple networks connect and exchange traffic [2]-[4].

2.6 Topology mapping in the African Context

Topology mapping within the African Context presents unique challenges and untapped opportunities, which largely inspired this project.

On the challenges front, topology mapping heavily relies on data from measurements. These measurements are initiated from devices called probes; this is not to be confused with the probes pertaining to packets sent to discover routes. In this context probes refer to vantage points. The term probe is commonly used in this context in RIPE ATLAS, which is popular platform for creating measurements and collecting data on computer networks [2]-[7]. The distribution of these probes is considerably skewed on the global scale and even more so within Africa. A significant concentration of these probes is in South Africa, with the remainder scattered across other parts of the continent. In fact, some nations lack probes altogether. This unequal distribution results in potential disparities in router and route discoveries [3][4][7]. Further complicating the landscape, some internet routes originating in Africa detour outside the continent, often touching down in other continents, before returning [5]

However, these challenges are not without opportunities. Firstly, there's a noticeable gap in research focused on mapping internet topologies within the African context. Moreover, existing literature that attempts the mapping, particularly at the router level, hasn't been updated in the recent decade. The latest attempt to map the internet topology at the continental scale was done by Gilmour et al [2007]. Given the dynamic nature of the internet, older literature might no longer provide an accurate picture of the current state of the topology [2]-[4]. Additionally, advancements in internet measurement methodologies, emergence of efficient graph databases, and advances in visualisation techniques and tools in recent years, presents as opportunity to produce a modern and arguably more responsive visualisations. Additionally, since existing research often delivers static topology maps, there exists an opportunity to develop a dynamic mapping system that continually gathers traceroute data, thereby offering a constantly updated representation of the African internet topology.

3. RELATED WORK:

Yang et al. [2016] introduced a topology visualisation tool called NetVizura to visualise the Network topology of National research and Education Networks (NREN) in Africa using traceroute data. This work adopted a user centred design approach where usability tests were conducted by recruiting human participants to assess the accuracy and the effectiveness of the visualisation. The visualisation shows traceroute information of all IP addresses to which traceroutes were sent on a map. The design of the visualisation used multiple traceroutes from different vantage points to achieve the mapping. It also displays the locations of the internet exchange points and the placement of fibre around Africa continent.

In the work by Gilmore et. al. [2007], uses the data collected from traceroute measurements and BGP datasets to generate router level and AS level maps. At router level, a java-based tool, Terrapix, was designed specifically to produce router level maps, where 2D and 3D geospatial visualisations were used to visualise the router and AS level topology. At AS level, Walrus tool created by CAIRA was used to display AS level structure of the African internet topology in a 3D space [2].

Munzner et al. [1996] developed a methodology to visualise the Mbone. The Internet MBone, or the Internet's multicast backbone, represents an efficient mechanism for transmitting data from a single sender to multiple receivers, minimizing packet replication [8]. This system is predominantly used for broadcasting real-time video and audio streams such as conferences and notable events. The MBone, like the Internet; grew exponentially with no central authority [8]. This unplanned surge has led to a topology that is suboptimal, garnering attention and concern from network providers and the multicast research community.

In response to the complexity of this growing structure, researchers innovated by visualizing the MBone's global topology []8. Through data visualisation techniques, the MBone's tunnel structure was mapped geographically onto a globe, with arcs indicating connections based on the MBone routers' latitudes and longitudes. The resulting visualisation was an interactive 3D representation, shared via the World Wide Web in VRML format. Such a visualisation makes understanding the structure more intuitive. The 3D visualisations are far more useful for people working in the MBone engineering process than the textual data. Moreover, it can serve as an educational medium for the general public [8].

4. ETHICAL, LEGAL AND PROFESSIONAL CONCERNS

4.1 Ethical Concerns

Human subjects were used during the usability testing. Any personal and sensitive data collected on them such as names, signatures and demographic data was not presented in the paper. Only aggregated data was presented and discussed.

Moreover, informed consent was exercised during usability testing. Users were made aware of what the study entails and what is expected of them during the testing procedure. They were then required to sign a consent form if they were interested in participating in this study.

Rewards were only handed out at the end of the study to minimize the risk of users’ comments and feedback being clouded by the rewards.

4.2 Legal Issues

A tool called ZMAP, was used to scan IP ranges to find reachable targets during the mapping phase of the project. For transparency purposes, it is important to disclose that internet-wide scans that are intended for exploiting networks are considered illegal. We only used this tool to discover targets [15]. We did not perform any exploitative activities on private networks nor attempted to access private resources.

5. USER-CENTERED APPROACH FOR REQUIREMENTS GATHERING AND PROTOTYPING

5.1 Gathering User Requirements and System Specifications

To ensure a robust design for our topology visualisation tool, we adopted a user-centred approach throughout our project. This involved direct interaction with potential users, understanding their needs, and tailoring our design to address those specific needs [9].

The process began with identifying the appropriate stakeholders for the project. By developing personas, a technique considered beneficial in prioritising product requirement and defining the scope of the problem that needs to be solved [9], we were able to identify the prospective users. Among these users were a university lecturer specializing in computer networks. Such an individual would predominantly use the topology visualisation tool in teaching. Additionally, we identified another key user category: a researcher in computer science, focusing on research in computer networks or internet systems. Researchers could use the tool either to gain insights into internet topology at the router level or as a foundation for further improvement. Lastly, we factored in senior and honours computer science students, especially those who completed a course in computer networks, who would use this tool to deepen their understanding of internet topologies.

A significant stakeholder in our project was our supervisor, a lecturer and researcher in the computer science department. He played a dual role: one, as the lecturer teaching computer networks and two, as a guiding force steering us towards the right direction. Through him, we connected with a Ph.D. student whose research interests align with computer networks. To achieve a more holistic understanding of user needs, we engaged with 5 senior computer science students and 5 honours students, who were selected randomly for the requirements gathering phase.

Our engagement began with structured meetings and interviews with our project supervisor and the Ph.D. student. Engagement with stakeholders in the form of meetings and interviews as prescribed by Chadia et al. [2004] serve as an effective tool in gathering data related to the needs and expectations of users. These sessions with stakeholders were geared towards understanding the core features the users envision for the topology visualisation tool. We predominantly asked open-ended queries to understand the expectations of the stakeholders in terms of what would make a minimum viable product (main use cases) and any prior experience with similar tools. Following this, we held similar discussions with the selected senior and honours students. All insights and requirements were noted and later transformed into use case diagrams.

5.1.1Gathered User Requirements

In our preliminary discussions with stakeholders, several primary user requirements emerged. First, stakeholders emphasized the need to visualize router topology with city-level accuracy, that is, ensuring that the visual representation of a router's location falls within the correct city boundaries. Another requirement is to visualize the links connecting routers, providing insight into the potential routes packets might traverse from one router to the next. Given the vast nature of router-level topology, it's essential for the visualisation tool to have an efficient filtering mechanism, enabling users to filter down to specific sections of the topology map. It's worth noting that these requirements are only an initial list, that is, the list of requirements collected is not exhaustive. They emerged from our first interactions with stakeholders. More requirements were continually gathered as the project advanced with each prototype iteration.

5.2 Prototyping

After gathering the initial requirements, and in line with the principles of participatory design, we moved forward with the creation of two low-fidelity, no-code prototypes that encapsulated these requirements [10]. This approach ensured active user involvement from the outset. We subsequently engaged with users, predominantly our project supervisor, with the goal of ensuring that our prototypes accurately match the requirements we had identified [10].

Opting for no-code prototypes was an intentional choice. The idea was to capture and understand user needs without delving into the intricacies of software development or committing to a particular technology stack prematurely. Rudd et al. [1996] highlight that such low-fidelity prototypes are invaluable during the requirements phase, serving as effective tools to capture, communicate and refine requirements. This is the rationale behind our decision to develop prototypes.

Upon receiving feedback and reaching consensus on the primary functionalities, we started developing the first iteration of our software prototypes.

5.2.1 Map-based Prototype

The first prototype focused on the visualisation of the router topology based on the map of the African continent. The visualisation consisted of the router level topology represented as a graph. The topology graph is then overlayed on the African map (see figure 5 in the appendix). The most common way of presenting computer networks is the use of a graph representation, which consists of nodes and edges or links [3][4][6]. Knight [2001], suggests that a visualisation is considered effective if it is suited for the task for which it supports, and the chosen representation metaphor matches the dataset. Fundamentally a router topology consists of interconnected routers, a graph representation is an appropriate and fitting choice for representing it.

In the overlayed topology graph, each router was represented by a unique marker. The placement of these markers reflects the geographical positions of the routers. This representation emphasized the spatial relationships between the routers, giving users an immediate sense of the distribution of routers. This approach resonates with the argument made by MacEachren et al. (1995), suggesting that geospatial visualisations grounded in real-world contexts enhance user comprehension of the visualisation.

In this prototype, users can interact with the map. When a marker is selected, a pop-up sticker appears above it, showing the name of its associated Autonomous System. Beside the map, there's an information panel that updates in real-time, providing details about the chosen router. This panel displays basic information like the Autonomous System name (e.g., SAIX-NET) and number (e.g., 5713), as well as more detailed data such as the city (e.g., Mthatha), country (e.g., South Africa), specific coordinates, IP address (e.g., 196.25.26.65), and broader region (e.g., EC for Eastern Cape). Ellis & Dix [2007] suggest that interactivity is crucial in the design of geospatial visualisations. Since the positions of the nodes are fixed to their geographical locations and can't be moved around, areas where large amounts of clustered points may occur. Making the router markets interactive eliminates the need to visualise the router information in plain sight, therefore minimising clutter. Screenshots of the prototype are included in the appendix (figure 5)

5.2.2 Tree Graph-based Prototype

For the second prototype, we opted for a conceptual shift, transitioning from the geographical framework of the map-based prototype to a tree graph format (see figure 6 in the appendix). At its core, this prototype was a force-directed graph diagram, where nodes were categorized and visually separated based on a hierarchical structure: countries, cities, suburbs, and individual routers.

The choice of a tree graph was informed by the need to provide a different perspective of the router topology — one that highlighted the inherent hierarchies in the geospatial placement of routers (routers are placed in cities, which are part of a region, the region is a country, and the country is in the African continent) [1] without being bound by geographical constraints. Contrary to the geospatial visualisation in the first prototype, the router nodes can be moved freely and therefore it theoretically feasible to visualise high level information such as the ASN number and name just next to the router nodes in plane sight. This makes it possible to immediately decipher where points of presence of specific ASes without interacting with the nodes. However, despite this level of detail, this tree graph visualisation had some inherent limitations. The most notable trade-off is the absence of the intuitive geographic clarity that the map-based visualisation provided, meaning that users had traverse down the tree to determine the geolocation of routers.

5.2.3 Initial Prototype Evaluation

We conducted an informal user study to assess the usability of the initial prototypes and verify if they sufficiently addressed core user requirements. At this stage of the development process, our goal was to gauge how well user requirements were captured.

The evaluation process involved setting up two separate meetings. Chadia e al. [2004] suggests engagement with stakeholders in the early stages of the design cycle could be used as a tool in evaluation of design alternatives, prototypes and the final artifact. The first meeting was arranged with the project supervisor and the second, with a select group of computer science students. During these meetings, both prototypes, along with their current functionalities, were demonstrated.

Feedback Received:

For the Map-based Prototype, the immediate feedback from our supervisor emphasized its potential, highlighting the overlay of the router network on the map as a commendable starting point. However, concerns were raised regarding the feasibility of representing individual routers due to the vastness of the router network and limited screen space. An insightful suggestion was to consider visualizing 'points of presence' rather than individual routers. In this context, a point of presence signifies a location where multiple routers of a specific Autonomous System are situated [4]. Consequently, if an Autonomous System possesses several routers in a particular city, only a single node or marker would be shown to represent all those routers. This approach would be a compact method to aggregate nodes [4]. Additionally, any links originally connected to routers of an Autonomous System in that specific locale would be rerouted to the point of presence symbolizing the AS's presence in that city.

In the case of the Tree graph-based prototype, the supervisor noted that while there are benefits associated with visualising the topology as a free-floating graph on an infinite canvas, it lacks the intuitive geographic representation evident in the map-based prototype. The map-based visualisation enables users to instantly discern a router's geographic location from its position on the map. In the tree graph, a user would need to navigate from the root nodes, traversing down to the leaf nodes, to determine router locations. Another notable critique was its inability to represent interconnections between routers situated in different cities.

5.2.4 Iteration One of Code Prototype

Following our initial prototype evaluations, we progressed to the next stage of our project: the development of the first software iteration of our prototype. The collective feedback from our supervisor and the user group demonstrated an inclination towards the map-based prototype. Considering the feedback, we decided to select the map-based prototype as a blueprint for the software version of the visualisation tool. Our development process was grounded in transforming the initial no-code prototype into the first working version of the tool (see figure 9 in the appendix).

5.2.5 Second Prototype Evaluation

After presenting the first software version of our prototype to the users, we received immediate feedback on the visualisation's crowded appearance. Trying to show every node and link at once made the interface look cluttered. This led to two main problems. First, nodes overlapped, making it hard for users to tell them apart. Second, the numerous links crisscrossing made it difficult to see individual connections clearly. This feedback underscored an important point: while it's essential to provide a comprehensive view, it's just as crucial to maintain clarity for an optimal user experience.

5.2.6 Iteration Two of Code Prototypes

Following feedback from the second prototype evaluation, we refined our prototype. Two primary issues were identified: Difficulty in distinguishing overlapped nodes and challenges in seeing connections between nodes.

Ellis & Dix [2007] highlighted a pertinent point relevant to our situation: in geospatial visualisations, where node positions are set by location, there's a risk of nodes or links overlapping. They suggested a range of clutter-reducing techniques, from visual adjustments to spatial changes and animations. Keeping that in mind, we added a zoom feature to our visualisation. This allows users to view the router topology in varying levels of detail. At a wide view, overlapping nodes combine into one, with their links converging. Originally, our nodes represented individual points of presence. But with our new method, when viewed from afar, nodes symbolize multiple points in a region. Zooming in separates these combined nodes back into individual points of presence, with their links reverting to their specific destinations (See figure 9 to 11 in the appendix).

For added user interactivity, we integrated a filtering system. This allows users to hone in on specific parts of the visualisation, thus reducing the clutter. Filters can be applied based on Country, Region City and Autonomous System

Lastly, to clear up any confusion about node connections, we enhanced link interactions. When a link is selected, the connecting nodes stand out, and a side panel provides in-depth information about the relationship between these nodes. (See figure 12 in the appendix).

5.2.7 Final Prototype Assessment

In line with our continuous evaluation approach, we organized a feedback session with our project supervisor and our chosen user group. During this session, we showcased the latest improvements made to our prototype and walked through all the developed use cases.

The primary feedback highlighted performance issues in the user interface. Users noted delays when performing actions such as zooming, dragging, and clicking. The time taken for the interface to load initially was also longer than anticipated. Addressing these performance concerns became our top priority for refining the prototype for final deployment.

5.2.8 Final Development Stage

The main aim for this stage was to address the performance delays in the interface. We thoroughly analysed our front-end architecture with the intention to identify the reasons for these delays.

Our deep dive revealed that the delays were caused by rendering too many objects, specifically node markers and their associated links. To enhance the system's speed, we made a change in how objects are rendered. Instead of displaying all links at once, we adjusted the system to show only the links of nodes visible in the current zoomed area. For example, when a user focuses on the Cape Town area, only links in that region are shown. This smart reduction in displayed objects greatly improved the system's speed, resulting in a smoother user experience.

6. SYSTEM DESIGN AND IMPLEMENTATION:

6.1 System design for the topology visualisation tool

In the development of the topology visualisation tool, a rigorous system design is essential to ensure the tool's reliability, efficiency, and scalability. The objective of this section is to present a clear architectural understanding of the system, specifically addressing how individual components interact and depend on one another. Our design process hinges on the principles laid out by Satzinger et al [2015] in their work, "Systems Analysis and Design in a Changing World".

In system design, diagrams play a pivotal role as they offer a concise and visual representation of complex processes and architectures. Diagrams, rather than verbose descriptions, enable stakeholders, to grasp intricate details quickly and unambiguously [13]. As Satzinger et al [2015] argues, this visual approach fosters clear communication, facilitates collaboration between stakeholders, and paves the way for more effective troubleshooting during subsequent phases. To leverage these advantages, we prioritized diagrammatic illustrations in our design strategy for the topology visualisation tool.

Our design framework for the topology visualisation tool is rooted in three primary diagrams. The Use Case Diagram highlights user interactions with the tool, setting clear a scope by capturing core use-cases [13]. The Package Diagram captures the tool's modular design which not only aids clarity during development but also simplifies maintenance and future improvements [13]. The State Machine Diagram, on the other hand, reveals the tool's behaviour by detailing system states and their potential transitions based on user actions or system events [13]. Together, these diagrams form a comprehensive blueprint ensuring the tool aligns with the requirements and envisioned use cases.

6.1.1 Use Case Diagram for the Topology Visualisation Tool

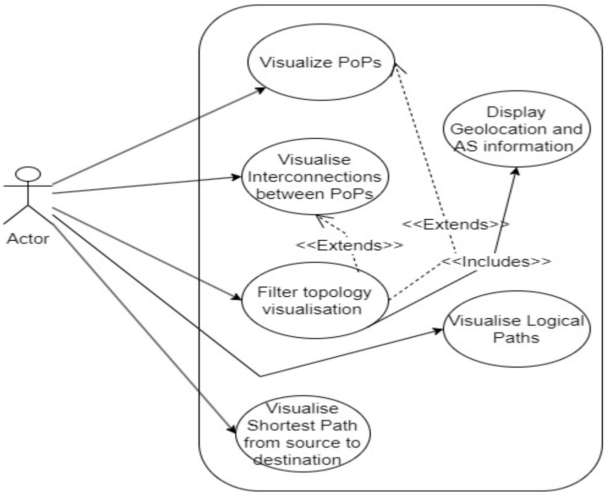


Figure 1 Use Cases

The use case diagram in figure represents all the user requirements that have been captured by the end of the system analysis phase. A subset of the use cases represented in the diagram were capture in the initial meeting with our users. The remaining use cases were captured incrementally during the prototyping and prototype evaluation phases (see Section 4).

6.1.2 Package Diagram of the Topology Visualisation Tool

The package diagram depicts all the system packages (components) and their dependencies. It depicts the high-level architecture of the system. The low-level implementation details are outlined in the system implementation section. The Measurement Package relies on the RIPE ATLAS platform for creating measurements, using the RIPE ATLAS API. It indirectly depends on the Probe and Target Info Management Package for data on probes and targets. The “Probe and Target Info Management” Package contains scripts for retrieving probe and target data. It's dependent on the RIPE ATLAS endpoint for probe data, the “AS Information” package for retrieving AS numbers, the Database packages for retrieving target IP addresses.

The GeoDatabase Package provides the geolocations of IP addresses, with MaxMind Geolite2 as the primary database and ip2location lite as a backup. The Autonomous System Information Management Package updates AS data, pulling from AFRINIC and RIPE. Measurement Ingestion pulls and processes RIPE Atlas measurements, then stores the measurements in a graph database, this database (neo4j community) is contained in the Database package. The Topology Visualisation Package manages UI logic, communicating with the Server Package to fetch and display router network data.

6.1.3 System State Diagram for Topology Visualisation Tool

To provide a clear understanding of the system's behaviour in response to user interactions, a system state diagram was employed (see figure 8 in the appendix). This diagram captures the various states of the topology visualisation tool transitions between these visual states as the user navigates through the interface.

Upon initial loading of the interface, the visualisation tool displays the default or "initial" state. In this state, an aggregate view of the router topology is presented. Nodes in this context symbolize groups of Points of Presence (PoPs) located within the same city, to minimise clutter.

When a user hovers over a node while in this default view, a pop-up emerges, revealing all PoPs represented by that particular node. A click on the node, on the other hand, pins the pop-up just above the node, exhibiting the AS names associated with the PoPs. Concurrently, all the corresponding PoP data is rendered in an information div adjacent to the map. This includes details such as Autonomous System names, ASN, City, Country Code, Country Name, IP, Latitude, Longitude, and Region.

Should the user zoom into a specific geographical area on the map, the tool undergoes a transition to what we term the "PoP" state. This transition is activated at a zoom level of 5.8, a metric determined empirically to ensure minimal overlap between nodes. Within this state, nodes evolve from representing clustered PoPs within a city to representing individual PoPs. Further enhancing the granularity of information in this state, links connecting nodes situated within the zoomed boundaries of the map are also visualized.

Users are afforded the option to filter the visualisations based on criteria like Country, City, Region, and Autonomous System. Two distinct filter modes cater to varied user objectives: the default filter mode and the highlight mode. In the default mode, only the PoPs matching the selected filters are rendered. Conversely, the highlight mode depicts all PoPs, but those matching the applied filters are highlighted. This approach recognizes that while some users might desire visualisation of specific PoPs exclusively, others might seek an understanding of interconnectivity between specific and other PoPs. Moreover, there is a feature that allows users to visualize a logical path extending from a source PoP to a designated destination PoP.

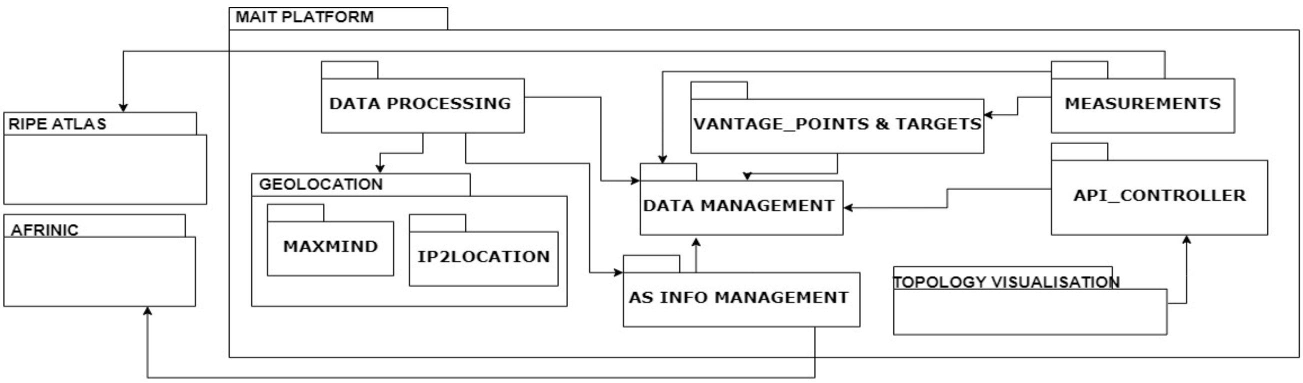


Figure 2 Package Diagram

6.2 System Implementation:

This section dives deeper into the lower-level implementation details. It outlines the system's structure, data processing methodologies, and the tools and frameworks employed.

6.2.1 System Architecture

We employed the 'pipe and filter' architectural paradigm. This model essentially embodies a flow wherein data is first harvested from various sources. Post-harvesting, this raw data undergoes a series of processing steps. The resultant processed data is permanently stored in the database [2]-[7][16][20]. Finally, this stored data is retrieved and translated into visual representations for users. Each phase of this pipeline feeds into the next. The system architecture diagram is included in the appendix (figure 7)

6.2.2 Network Discovery (Router Discovery):

To obtain router IP addresses, periodic traceroute measurements will be performed with the help of the RIPE Atlas API. The measurements are run from multiple vantage points to discover as many routers as possible and ensure high coverage of the African internet topology [2]-[7].

In the context of this project, a vantage point refers to the point from which measurements are initiated. We are leveraging RIPE ATLAS probes as our vantage points. That is, measurement contain a probe and target pair. The traceroute measurements are run from the source to the target. RIPE ATLAS has a collection of probes strategically placed across the continent [5]. Probes are network devices from which a measurement could be initiated [5]. Probes are essentially used as vantage points. We pulled the probe data from RIPE atlas; this includes IDs and IP addresses of all the probes active probes in Africa.

The next step involved fetching target IP addresses [2]-[5]. To Achieve this, we wrote a script that pulls Autonomous System Numbers from the internet registry for Africa, AFRINIC. The next step in the pipeline involved pulling the prefixes associated with each Autonomous System Number from RIPE stat. The next the step in the pipeline involved running a Zmap scan on the prefixes to determine reachable targets with the IP ranges. Scanning IP range is an expensive operation [15]. To minimise the time taken to scan all the prefixes, we decided to run the scans in parallel. Multiple processes are spawned, each of them scans n IP ranges. With the probe and target data fetched, the next objective was to generate traceroute measurements from each probe to every target, ensuring high route coverage [2]-[5]. We collected a total of 217 probes and 6,163 targets. We wrote a script that periodically hits the RIPE ATLAS end point and systematically creates traceroute measurements from each vantage point to every other target, that is, there is a measurement created for each source(probe) and destination (target) pair. The RIPE ATLAS tool leverages the ICMP protocol for traceroute measurements [4]. The RIPE ATLAS endpoints return a unique ID for every successfully created measurement, IDs are then stored in measurement table in the database.

The next step in the pipeline involves pulling the measurement data from RIPE ATLAS. Given that we already stored the measurement IDs of the successfully created measurements, we wrote a script that periodically check the database for successfully created measurements, for each measurement, a JSON payload containing the measurement data is returned. The measurement data returned includes the IP addresses of each hop (router), Time to Live metric per hop, Round Trip Time metric per hop [4].

6.2.3 Measurement Ingestion: building the router topology.

The ingestion process plays a pivotal role in transferring the traceroute measurement data from the RIPE ATLAS into our Neo4j database, aiming to build a router graph that visually portrays the interconnectedness of routers. As previously mentioned in the background sections, topologies are commonly represented as graphs [4].

Each Measurement consist of a series of hops which represent the path taken by packets from the vantage point to the target. Each hop in the measurements represents a router. Given we already stored the IDs of the measurements created, the first step in ingesting the measurements in the graph database involved pulling the measurement results from RIPE ATLAS

The core of the process commences which involves processing each individual traceroute measurement. The emphasis here is on understanding the network hops associated with each measurement. From each network hop, the associated router IP (denoted in the data as the 'from' field) is identified. If this router's IP doesn't already exist in our database, it is added as a unique node [2]-[4][7].

The data's inherent structure often means that for every router in a traceroute measurement, there is a preceding router from the previous hop. Thus, for each newly identified router (or node), a direct relationship is established with the router associated preceding hop. This relationship forms an edge in our graph, illustrating the direct connectivity between the two routers. It's through this step-by-step addition of nodes and edges that the broader router graph takes shape, providing a visual representation of router interconnectivity based on the traceroute measurements [2]-[4][7].

Post-processing, the system marks the measurement ID of the traceroute measurements that have been ingested as 'read' in the database. This ensures that in subsequent operations, the same data isn't redundantly processed again.

The choice of Neo4j as the database for this project was intentional. Neo4j is a leading graph database specifically designed to efficiently manage highly interconnected data. In the realm of network topologies, where understanding relationships between routers is important, the graph data model used in Neo4j is suitable. Each router and its interconnections can be naturally represented as nodes and edges, making Neo4j a clear choice for our use case. Moreover, Neo4j's inherent capabilities in pattern detection, shortest path identification, and its flexibility in querying complex relationships provide substantial advantages over traditional relational databases. This empowers us to derive deeper insights into the intricate network topology and the relationships between routers.

To summarize, the measurement ingestion mechanism operates in a structured manner, channelling raw traceroute data from the RIPE Atlas platform and then transforming it into a structured topology graph in our Neo4j database. The choice of Neo4j underscores our commitment to utilizing tools that offer the best alignment with our objectives and the nature of our data.

6.2.4 Router Geolocation and Autonomous System (AS) Information Ingestion:

Post the ingestion of router data into our Neo4j database, the next phase in the data pipeline involves geolocating these routers and pulling relevant Autonomous System (AS) information. That is, each router node is updated with geolocation and AS information.

6.2.5 Primary Geolocation Methodology – MaxMind GeoLite2:

Our pipeline's backbone for router geolocation primarily leans on the reputable MaxMind GeoLite2 database [30]. This extensive database offers granular geolocation details, from country codes and city names to latitude-longitude coordinates at city level accuracy, all based on IP addresses. Furthermore, it offers key AS data, providing insights into which autonomous systems each router falls under.

For optimal efficiency and to keep costs minimal, both the GeoLite2 City and ASN datasets are locally hosted [30]. This design choice primarily stems from financial considerations. Local hosting eradicates the overhead associated with frequent cloud-based API calls, which can rapidly accumulate, especially given the sheer volume of our data.

However, this approach does introduce a notable challenge. Local datasets, while immensely cost-effective, might not be as up to date as their cloud-hosted counterparts. This could lead to slightly reduced accuracy [30].

Operational Flow:

The system initiates its workflow by fetching router IP addresses from the database, these IP addresses are then geolocated. To determine geolocation, the system utilizes the locally hosted MaxMind GeoLite2 database [30]. The pipeline is equipped with a fallback mechanism. Should MaxMind fail to geolocate an IP address the system switched to the IP2Location database, another locally hosted geolocation database [31]. This fallback mechanism ensures robustness.

6.2.6 Data Processing to create a point of presence network:

During the requirements collection phase, the visualisation of router network on a map-based interface proved challenging due to the density of routers and restricted visualisation space. Representing each router individually, while precise, resulted in a crowded interface that could be overwhelming for users. The feedback we received emphasized the need for a more aggregated visualisation. See Section 4 for more detail.

6.2.7 Requirement-driven Solution: Points of Presence (PoP):

Our supervisor suggested aggregating the topology into 'points of presence' (PoP). Within this scope, a PoP signifies a location where numerous routers from a singular Autonomous System (AS) are consolidated. This means that, instead of visualising the individual routers in a city, we aggregate routers belonging to the same AS and are a located in the same city into a single PoP node [4].

6.2.8 Data Processing Approach:

1. Initial Data Structure – The Router Graph:

Initially, the router topology represented as a graph, was structured to directly mirror the measurement data. Each node corresponded to a unique router. While this one-to-one representation was accurate, it created clutter in the visualisation, as previously discussed.

2. Cloning for Data Integrity:

Before restructuring the topology graph. we cloned it, resulting in a 'RouterClone' graph. This is a clone of all nodes (routers) and their links, safeguarding the original dataset's integrity.

3. Data Aggregation Strategy:

- Merging Process: Routers were merged into a PoP node. The PoP nodes represented a collection of all routers belonging to a singular AS in a city. For instance, all routers in Cape Town belonging to AS 'XYZ' would be visually represented by a single PoP node denoting AS 'XYZ's presence in the city.

- Re-routing Links: Post-aggregation, it was crucial to sustain the overarching network structure. All links, both incoming and outgoing, previously linked to the merged routers, were redirected to the corresponding PoP nodes. This process ensured that the topology's integrity remained intact in the PoP-focused graph.

4. Performance gains:

Over and above reducing clutter, reducing the node and relationship count by merging the nodes, reduced the computational load on the client side. This led to better responsiveness of the visualisation.

Additionally, the PoP level abstraction eliminates the need for alias resolution. Alias resolution primarily addresses the challenge of identifying multiple IP addresses that belong to the same router [4][6]. At the PoP (Point of Presence) level, the focus shifts from individual routers to a higher-level aggregation.

6.2.9 Data Processing for Optimized Topology Visualisation in the Front-end:

In the preceding stage of our data pipeline, we converted the router graph into the PoP graph. As highlighted in the requirements analysis section, there was a challenge tied to our approach for geolocation. The geolocation databases we hosted locally only offer city-level accuracy for latitude and longitude coordinates. Consequently, PoPs within the same city would overlap in the visualisation.

To address the overlapping nodes, we adopted a strategy to spatially distribute PoPs located in the same city. We wrote a script that add random noise to the geographical coordinates of the PoP nodes. This "jittering" ensures that each PoP appears distinct on the map while ensuring that the resulting geolocations are within a 50km radius from the original location. The utility of this approach is evident when visualizing interconnections between PoPs within the same city. Previously, we could not visualise the interconnections because they were hidden by the overlap between the nodes.

6.3 System Development Methodology for the Topology Visualisation Tool

We employed the agile methodology for the software development, addressing the complexities of visualizing network topologies and adapting to user needs. Our process involved organizing sprints that lasted two weeks, prioritizing tasks based on a product backlog. This iterative method promoted ongoing enhancement by integrating feedback from users and the project supervisor [16][17][20].

Early prototyping helped us create initial versions of the tool, which allowed for user interaction and feedback. This helped us address visualisation challenges and refine features like marker clarity and zoom functions. Regular meetings with the project supervisor after each sprint confirmed our progress and ensured alignment with the project's goals. Incorporating agile methods, prototyping, and consistent feedback was essential in developing a tool that balanced technical functionality with user-focused design [16][17][20].

6.4 Testing

Throughout the development of the topology visualisation tool, we tested our system components for reliability and accuracy, including API, unit, and integration tests.

6.4.1 API Testing

API testing was essential to confirm that endpoints functioned correctly and returned the right data [18]. We used Postman for this purpose [18]. Our main objective was to ensure consistency in our API results. For instance, the AFRINIC API provided a list of ASNs in Africa, which we stored in the database. Meanwhile, the Ripe API delivered AS-related prefixes, and the RIPE API for Probes produced IP addresses and probe IDs for African probes. The Ripe Atlas APIs were also crucial for creating and retrieving traceroute measurements. Further details, including test cases and results, are in the appendix.

6.4.2 Unit and Integration Testing

Our unit tests focused on examining each function individually to ensure it performed as intended. Integration tests, on the other hand, looked at how these functions worked together. Specifically, we checked the data flow between the Neo4j database, various endpoints, and the scripts, confirming that the system as a whole functioned properly while maintaining data integrity [19]. A comprehensive table detailing the test cases, inputs, expected outputs, and actual results is available in the appendix. Through our testing procedures, we aimed to establish the dependability of our data processing methods [19].

6.5 Maintainability and Portability

6.5.1 Design Patterns

The development of the Topology Visualisation Tool was guided by standard software design patterns to enhance modularity, maintainability, and efficient problem-solving. We segmented our system's architecture into distinct components, with each focusing on a particular part of the data pipeline [20]. We utilized the Componentization Pattern, dividing the software into separate components, each dedicated to a specific functionality. The architecture also adheres to the single responsibility principle, that is, scripts are designed for distinct tasks like creating, storing, and managing measurement data [20].

Additionally, we adopted the Low Coupling Design Pattern, ensuring minimal interdependence between scripts [20]. Each script operates independently, only communicating through set interfaces. This approach not only allows for component interchangeability but also boosts the system's resilience against issues. These design decisions enhance the robustness and adaptability of the system.

6.5.2 Portability

Our topology visualisation tool was initially developed on a local Linux machine (Ubuntu), and utilized local resources including the neo4j community database, ip2location, and MaxMind databases. Built as a NodeJs web application, the frontend was designed using standard technologies such as CSS, HTML, and Javascript. The MapBox API was incorporated for mapping capabilities. The backend, including the express server and data pipeline functions like data pulling, processing, and ingestion, was entirely developed in JavaScript. Considering the local nature of the initial development, there was a need to address the tool's portability to ensure it could be easily transferred and run on different systems or platforms.

For wider accessibility and real-world application, we decided to deploy the tool on a remote server. The initial remote deployment was done on Heroku, which required transferring both the source code and the database to the cloud. The source code was deployed using the Heroku CLI, integrated with git for easier version control and updates. The database transition involved migrating from the locally hosted neo4j community database to AuraDB, the cloud version of neo4j. After migration, adjustments were made in the codebase to connect to the cloud-hosted database. While the application was developed using Linux-specific shell scripts for automation, it's designed to also run on Windows servers. To achieve this, the Linux .sh scripts would need to be refactored into Windows-compatible .cmd scripts. Package management across different platforms is handled through npm, with dependencies clearly defined in the package file.

The comprehensive deployment guideline is documented in the README.MD file (see the folder containing the source code).

7. **VALIDATION**:

Our goal was to develop to a visualisation tool for the internet topology at the router level. This tool's primary purpose was to clearly display the intra-continental topology, identify individual routers, show their geographical locations, the links between them, the logical pathways spanning regions and countries, and finally, highlight the ASes managing this infrastructure.

Building the topology graph involved processing measurements data. Further processing of the resulting topology graph was required to so that is it is suitable for visualisation in a user interface. To ensure our tool was accurate and reliable, we conducted a validation study. We aimed to determine whether the topology visualisation tool accurately represent the African router level topology.

We engaged with the PhD student we recruited during the requirements gathering phase to assess the tool. Their research interest lies in computer networks, and they have four years of research experience in the field. As part of our methodology for the validation study, we comprehensively explained how we gathered, processed, and integrated the data into our visualisation, ensuring he understood both the technical and visual elements of our tool. This process is well explained in the System design and implementation section of the paper.

The expert assessed the following:

1. Accuracy: He checked the correctness of our data transformations and aggregations. The feedback was that our tool-maintained data integrity.

2. Representation Accuracy: He compared the tool's visual output with what was expected from a router-level internet topology visualisation for Africa. He confirmed that the visual representation was as expected.

3. Clarity: The expert evaluated if the nodes, labels, and connections in the visualisation were easily distinguishable. His feedback indicated that the visual elements were clear.

4. Responsiveness: He tested the speed of the tool in terms of loading and interaction times. He confirmed that the tool was quick and responsive.

5. Functionality: All interactive features of the tool, such as zoom, pan, and click actions, were tested. The expert found them to be working smoothly.

6. Feedback Mechanisms: Finally, he assessed if the system provided clear and useful feedback to user actions. He confirmed that it did.

In summary, the expert's assessment of our visualisation tool was positive, indicating that it met the criteria set for accuracy, clarity, and user experience.

8. USABILITY TESTING:

One of the aims for the project is to develop a usable visualisation by leveraging user centred design techniques. We conducted usability tests to evaluate how well the final topology visualisation tool meets this objective. We adopted a mixed method study for the usability testing, that is, we are collected both qualitative and quantitative data pertaining to user behaviours, preferences, experiences, and interactions with the topology visualisation tool [10][25][26].

Additionally, we adopted system usability scoring to supplement the usability testing. Bangor et al. [2008] in their work “An Empirical Evaluation of the System Usability Scale” concluded that System Usability Scoring is a valuable and robust tool in assessing the usability of a broad spectrum of user interfaces.

8.1 Participant Selection

We already identified and recruited a group of prospective users during the requirements gathering phase through the use of personas [9]. The project supervisor played an important role in assessing the usability of the tool during the development process and steering us in the right direction. During this process they played the role of lecturer teaching computer networks and an expert in computer networks research. For the final usability testing we also recruited 10 senior and 5 honours computer science students.

8.2 Quantitative Data Collection

In the qualitative study, users were provided with a set of tasks to execute on the topology visualisation tool. Their success rate and the time taken to complete each task were recorded [10]. The aim was to determine whether the tool intuitively communicates the router level topology and is user friendly.

While working on these tasks, users were instructed to verbally express their thought processes. This method, known as "think aloud heuristics", allowed us to gauge the success rate of users and intervene or assist if they seemed to be on the wrong track [10].

Below is a selection of tasks from the quantitative study. The comprehensive list can be found in the appendix under the specified table.

1. Zoom Functionality & Aggregate Views

- Activity: Zoom out fully to view aggregate data representation.
- Question: How many aggregate markers or POP groups can you spot on the map?

2. Complex Information Retrieval

- Activity: Locate all points of presence for ASN: 2018.
- Question: Identify the Autonomous System POP linked between Durban and Bloemfontein.

Success for each activity was determined by whether the user could correctly complete it the activity with minimal assistance.

8.3 Qualitative Data Collection

The qualitative study was more focused on understanding the experiences, perceptions, and preferences of the target users when interacting with the internet topology visualisation tool. The qualitative study would focus on gaining insights into how users make sense of the tool, how it meets their needs, and any challenges they face while using it [10][25].

Beyond numbers and metrics, the subjective experience and sentiments of users interacting with our topology visualisation tool offer invaluable insights that structured qualitative data does not capture [10][25]. This qualitative phase aims to encapsulate the nuances of user emotions, perceptions, and preferences.

To collect qualitative data, we asked the selected users open ended questions after completing the user tasks given to them. Additionally, while users performed these tasks we observed and recorded any general comments and errors they made [10].

9. RESULTS AND DISCUSSION:

The usability testing results for the topology visualisation tool, encompassing both quantitative and qualitative aspects, offer a comprehensive understanding of the tool's overall effectiveness and areas that require improvement. From a broad perspective, the tool exhibits strong usability traits, this is reflected by the high completion rates for most tasks. However, specific insights from the qualitative feedback help highlight areas that contributed to less-than-perfect scores.

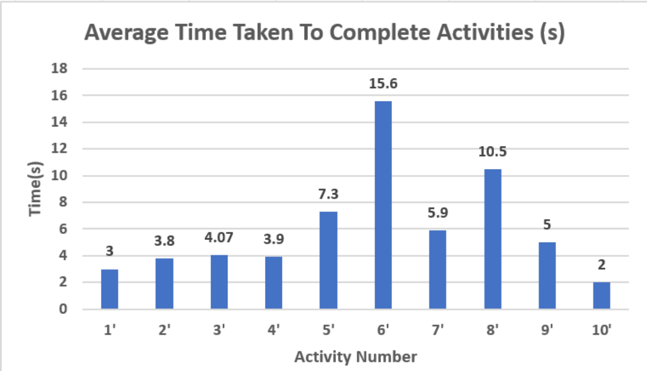


Figure 3 Average Time to Complete Activities

9.1. Primary Interactions:

The activities that focused on essential features of the tool, such as identifying PoPs and aggregate markers, achieved 100% success. The point was to test the user would be able to determine the names of the autonomous systems that have points of presence in the area on the map where the marker is pinned. The qualitative response we received from users is that performing the activity was "intuitive" and "simple". All it takes to perform this activity is to hover on the makers. The points of presence represented by the maker will then show in a pop-up above the marker.

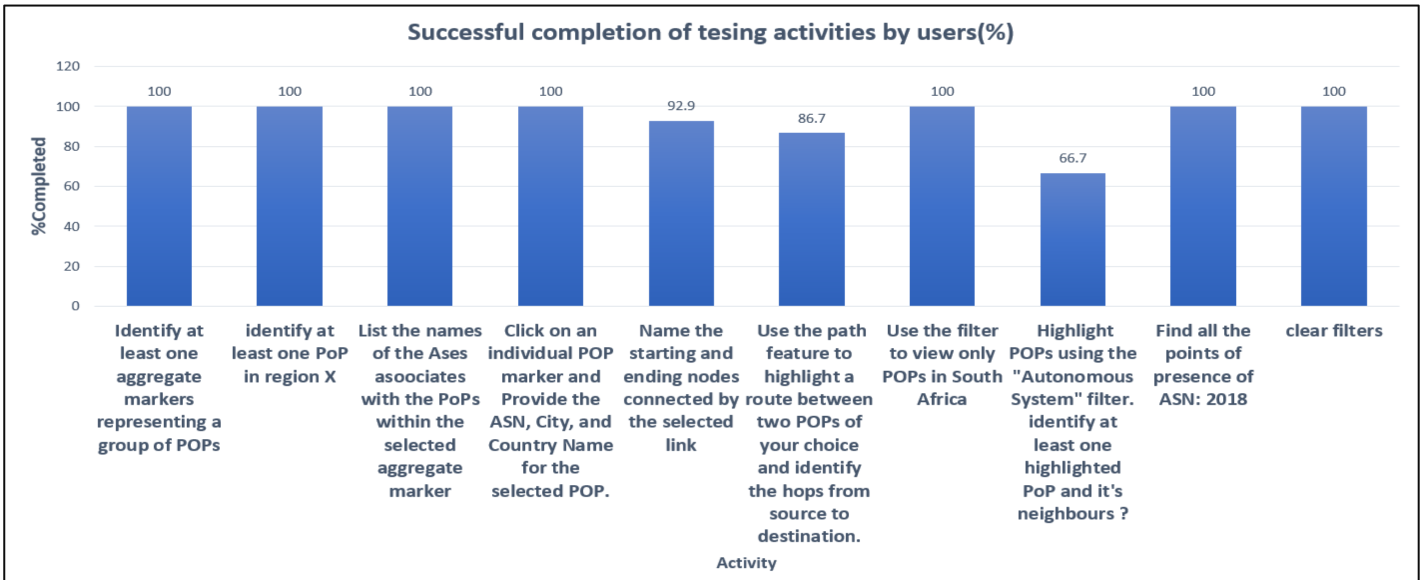


Figure 4 Successful completions of activities by users

9.2. Activities that required complex interaction:

The path feature, which highlights the shortest path from source PoP to destination PoP, showed a success rate of 86.7%. While it is still a high success rate, its longer average completion time and reduced success percentage in comparison to other tasks suggests added complexity. The idea behind this activity was to determine whether the user will be able to identify the shortest path from a source pop to a destination PoP using the tool. The qualitative feedback provides clarity on this: One of the users who failed to complete the task mentioned that using this feature involved a lengthy process. keeping in mind that you have to click on the source text box then select the source PoP/node and do the same for the destination PoP was difficult.

Apart from the path feature, users mentioned elements like the clickable links and highlighting features were not intuitive. The clickable links feature show the PoPs connected by the link once the user clicks on it. The user had to switch to clickable link mode for this feature to work. Similarly, using this feature involved too many steps.

9.3. Interface Elements and Labelling:

The "Highlight PoPs belonging to Autonomous System of your choice" task, boasts a 66.7% completion rate, was the lowest among all tasks. This task was meant to test the usability of the Highlight filter feature, which highlights all the nodes that match the filter values. Qualitative feedback consistently pointed towards specific UI elements as sources of confusion. The labelling of the toggle button "HLT" was not obvious "Highlight" would have been a better label. Such interface ambiguities can hinder task execution, even if the underlying feature is designed well. The importance of clear labeling and intuitive UI elements becomes evident in such cases, as they can significantly impact task completion rates.

9.4. Comparative Context and Expectations:

Several participants compared the tool to other mapping platforms, drawing attention to familiar features they appreciate (e.g., zoom-in revealing more detail) and minor issues like poorly labeled buttons. Such feedback, when juxtaposed against the quantitative data, offers insights into areas where the tool meets industry standards and where it may deviate.

9.5 System Usability Scale Method

SUS Question	score
I think that I would like to use this system frequently	**4.5**
I found the system unnecessarily complex	**1.6**
I thought the system was easy to use	**4.6**
I think that I would need the support of a technical person to be able to use this system	**1.5**
I found that the various functions in this system were well integrated	**4.5**
I thought that there was too much inconsistency in this system	**1.6**
I would imaging that most people would learn to use this system very quickly	**4.7**
I found the system very cumbersome to use	**1.4**
I felt very confident using the system	**4.6**
I needed to learn a lot of things before I could get going with this system	**1.5**

Table 1 System Usability Score

The System Usability Scale (SUS) is a method introduced by John Brooke in 1986 for evaluating perceived usability through a 10-item questionnaire with answers ranging from 1 to 5. To calculate the overall usability score, Ginanjar Wiro Sasmito et al. propose distinct scoring steps for even and odd questions. Specifically, odd questions have their value reduced by 1, while for even questions, the score is determined by subtracting the participant's response from 5. After aggregating the results, the total is multiplied by 2.5 to derive the final score, capped at a maximum of 100 [29].

Results for the Topology Visualisation Tool (see Table 1):

For odd-numbered questions:

[Odd Total = (4.58 - 1) = 3.58]

For even-numbered questions:

[Even Total = (5 - 1.52) = 3.48]

Combining both results:

[Combined = 3.58 + 3.48 = 7.06]

[SUS Score = 7.06 times 2.5 = 17.65 times 4 = 70.6]

With a score of 70.6, our topology visualisation tool boasts good usability. This suggests that the tool is relatively intuitive and aligns with user expectations. However, there remains room for improvement. Drawing insights from the methodology presented by Sasmito et al., we can infer that our tool provides good user experience but can benefit from further refinement [29].

10. CONCLUSION:

This project set out with the intent to develop a topology visualisation tool that accurately mapped the African router level topology and is user-friendly. Leveraging user-centred design techniques to ensure that the final tool met the requirements of the users and is usable. Given that the mapping the African router level topology required the collection and processing of measurement data, it was necessary to perform a study to validate whether the approach used to process the data was correct and the resulting topology visualisation was accurate.

We involved a PhD student with expertise in computer networks to assess our tool. This expert examined various aspects, such as the integrity of the data after processing, the accuracy of the visual representation, functionality, and user interaction. In conclusion, the expert's assessment validates that our tool effectively and accurately visualizes the African router-level internet topology. However, some areas, such as efficiency of the algorithms used for data processing were not validated, suggesting the need for a closer examination.

We then conducted a mixed-method study to assess the usability of the tool. The quantitative data revealed a commendable performance in primary tasks, signalling the tool's robustness in its foundational capabilities. The qualitative insights, on the other hand, enriched our understanding, highlighting user challenges with certain advanced features, thus pointing towards avenues of improvement.

Adding another layer of assessment, the System Usability Scale (SUS) awarded our tool a score of 70.6. While this is a favourable score that suggests the tool is usable, there is, as always, room for improvement. The gaps identified in the usability tests, particularly those highlighted in the qualitative feedback, provide direction for this improvement.

The feedback concerning interface elements, labelling, and user comparisons with other familiar tools indicates that with careful iteration, the tool's usability can be enhanced further.

This project has indeed laid the foundation for a competent visualisation tool. However, usability is an ever-evolving metric. Keeping the lines of feedback open with users will ensure that the tool remains relevant, user-friendly, and technically accurate. The principles laid out by Bangor et al. (2008) underscore the importance of continuous usability assessment across diverse interfaces. In that spirit, our project not only stands as a testament to our current achievements but also as a roadmap for future refinements.

11. LIMITATIONS AND FUTURE WORK

This project encountered several limitations. Firstly, we relied on a locally hosted geolocation database. These databases are the lite versions of the cloud hosted geolocations databases. The pool of IP addresses they are able to geolocate is not as wide and the precision is lower. There were instances where we were not able to determine the name of the Autonomous Systems associated with specific IP addresses because the geolocation databases default to AFRINIC (The internet registry for African). The geolocation databases do not offer the same precision as cloud-hosted versions. Secondly, while the tool can determine the shortest path between two points, it does not provide data on Round Trip Times.

We used RIPE ATLAS to create traceroute measurements. The Distribution of RIPE ATLAS probes across the continent is uneven. Most ripe Atlas probes are hosted in South Africa and the rest are distributed across the continent [27]. Other countries do not have any probes at all. This uneven distribution of probes across the continent raises concerns about potentially undiscovered routes and routers, which could affect the tool's coverage of the router level topology.

Another limitation we faced is that when visualising densely populated portions of the router topology, the user interface lags. Although this is not a major issue, it is still noticeable. The lag is due to the increase in the number of objects (links and routers) the visualisation engine (MapBox) has to manage. Future improvements could include using a more efficient visualisation engine or optimising the visualisation for responsiveness.

12. REFERENCES

[1] James F. Kurose and Keith W. Ross. 2017. *Computer Networking (7th ed)*. Pearson.

[2] Gilmore, J., Huysamen, N., & Krzesinski, A. 2007. Mapping the african internet. In Proceedings Southern African Telecommunication Networks and Applications Conference (SATNAC), Mauritius

[3]. Benoit Donnet and Timur Friedman. 2007. Internet topology discovery: a survey. *IEEE Communications Surveys & Tutorials* 9, 4: 56–69. https://doi.org/10.1109/comst.2007.4444750

[4] Reza Motamedi, Reza Rejaie, and Walter Willinger. 2015. A Survey of Techniques for Internet Topology Discovery. *IEEE Communications Surveys & Tutorials* 17, 2: 1044–1065. https://doi.org/10.1109/comst.2014.2376520

[5] Chantal Yang, Hussein Suleman, and Josiah Chavula. 2016. A topology visualisation tool for National Research and Education Networks in Africa. 2016 IST-Africa Week Conference. https://doi.org/10.1109/istafrica.2016.7530605

[6] Ken Keys. 2010. Internet-scale IP alias resolution techniques. *ACM SIGCOMM Computer Communication Review* 40, 1: 50–55. https://doi.org/10.1145/1672308.1672318

[7] Staff, R. N. 2015. Ripe atlas: A global internet measurement network. *Internet Protocol Journal*, *18*(3), 2-26.

[8] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner. 1996. Visualizing the global topology of the MBone. *Proceedings IEEE Symposium on Information Visualisation '96*. https://doi.org/10.1109/infvis.1996.559226

[9] Tomasz Miaskiewicz and Kenneth A. Kozar. 2011. Personas and user-centered design: How can personas benefit product design processes? *Design Studies* 32, 5: 417–430. https://doi.org/10.1016/j.destud.2011.03.003

[10] Abras Chadia, Diane Diane Maloney-Krichmar, and Jenny Preece. 2004. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction* 37, 4: 445–456.

[11].Jim Rudd, Ken Stern, and Scott Isensee. 1996. Low vs. high-fidelity prototyping debate. *Interactions* 3, 1: 76–85. https://doi.org/10.1145/223500.223514

[12] Knight, C., 2001. Visualisation effectiveness. 2001 Int. Conf. Imaging Sci. Syst. Technol. (CISST 2001) , Cisst (2001), 249–254.

[13] John W. Satzinger, Robert B. Jackson, and Stephen D. Burd. 2015. *Systems Analysis and Design in a Changing World*. Cengage Learning.

[14] Ellis, G. & Dix, A. 2006. An Explorative Analysis of User Evaluation Studies in Information Visualisation. Proceedings of the 2006 AVI workshop on BEyond time and errors novel evaluation methods for information visualisation BELIV 06: 1. http://doi.org/10.1145/1168149.1168152

[15] Durumeric, Zakir, Eric Wustrow, and J. Alex Halderman. 2013. {ZMap}: fast internet-wide scanning and its security applications.In *22nd USENIX Security Symposium (USENIX Security 13)*, pp. 605-620.

[16] Günther Ruhe and Claes Wohlin. 2014. *Software Project Management in a Changing World*. Springer.

[17] M. Coram and S. Bohner. The Impact of Agile Methods on Software Project Management. *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*. https://doi.org/10.1109/ecbs.2005.68

[18] Pratik Prakash Kore, Mayuresh Jaywant Lohar, Madhusudan Tanaji Surve, and Snehal Jadhav. 2022. API Testing Using Postman Tool. *International Journal for Research in Applied Science and Engineering Technology* 10, 12: 841–843. https://doi.org/10.22214/ijraset.2022.48030

[19] A. M. R. Vincenzi, J. C. Maldonado, E. F. Barbosa, and M. E. Delamaro. 2001. Unit and integration testing strategies for C programs using mutation. *Software Testing, Verification and Reliability* 11, 4: 249–268. https://doi.org/10.1002/stvr.242

[20] Ian Sommerville. 2015. *Software Engineering, Global Edition*.

[21] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* 24, 6: 574–594. https://doi.org/10.1080/10447310802205776

[22] Hongcheng Huang and Ziyu Dong. 2013. Research on architecture and query performance based on distributed graph database Neo4j. *2013 3rd International Conference on Consumer Electronics, Communications and Networks*. https://doi.org/10.1109/cecnet.2013.6703387

[23] Mario Miler, Damir Medak, and Dražen Odobašić. 2014. The shortest path algorithm performance comparison in graph and relational database on a transportation network. *PROMET -*

Traffic&Transportation 26, 1: 75–82. https://doi.org/10.7307/ptt.v26i1.1268

[24] Garima Jaiswal. 2013. Comparative analysis of Relational and Graph databases. *IOSR Journal of Engineering* 03, 08: 25–27. https://doi.org/10.9790/3021-03822527

[25] Gail D. Caruth. 2013. Demystifying Mixed Methods Research Design: A Review of the Literature. *Mevlana International Journal of Education* 3, 2: 112–122. https://doi.org/10.13054/mije.13.35.3.2

[26] Courtney A. McKim. 2016. The Value of Mixed Methods Research. *Journal of Mixed Methods Research* 11, 2: 202–222. https://doi.org/10.1177/1558689815607096

[27] 2014. Distribution of RIPE Atlas Probes. *RIPE Labs*. Retrieved from https://labs.ripe.net/author/emileaben/distribution-of-ripe-atlas-probes/

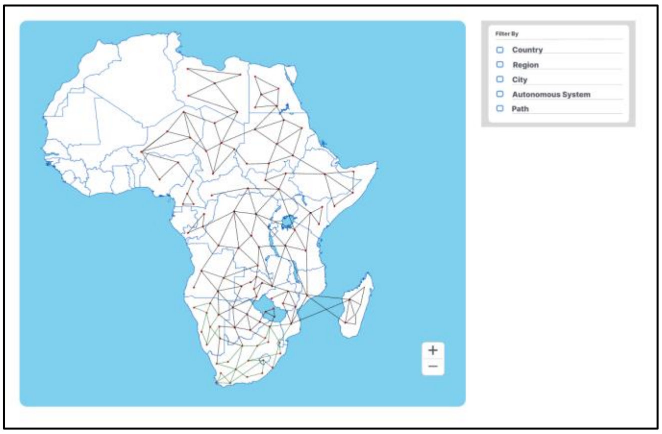
[28] Brooke, J., 2013. SUS: a retrospective. *Journal of usability studies*, *8*(2), pp.29-40.

[29] Ginanjar Wiro Sasmito, La Ode Mohamad Zulfiqar, and M. Nishom. 2019. Usability Testing based on System Usability Scale and Net Promoter Score. *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. https://doi.org/10.1109/isriti48646.2019.9034666

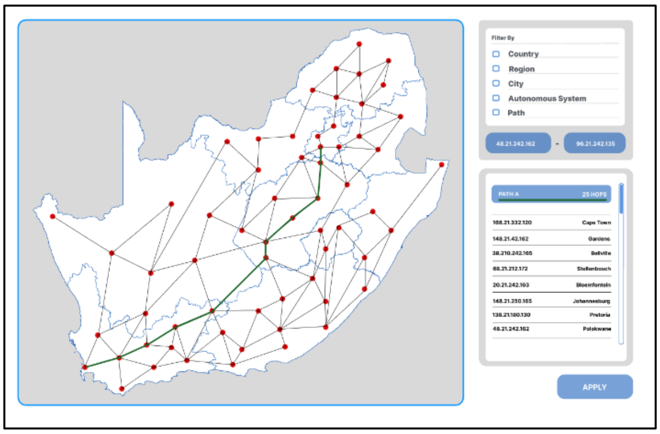
[30] MaxMind Server IP Addresses. *MaxMind Developer Portal*. Retrieved from https://dev.maxmind.com/geoip/geolite2-free-geolocation-data.

[31] IP2Location. IP2Location LITE Database. *IP2Location*. Retrieved from https://lite.ip2location.com/ip2location-lite?lang=en_US.

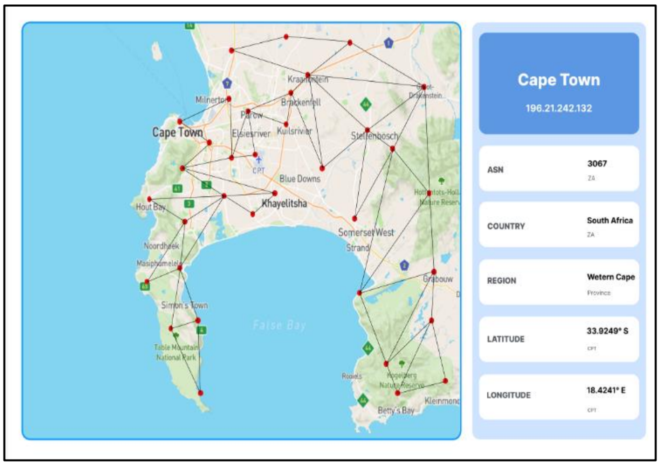
13. Appendix



a) Landing screen



b) Showing Path from source to destination



c) When Router Marker clicked

Figure 5 map-based prototype.

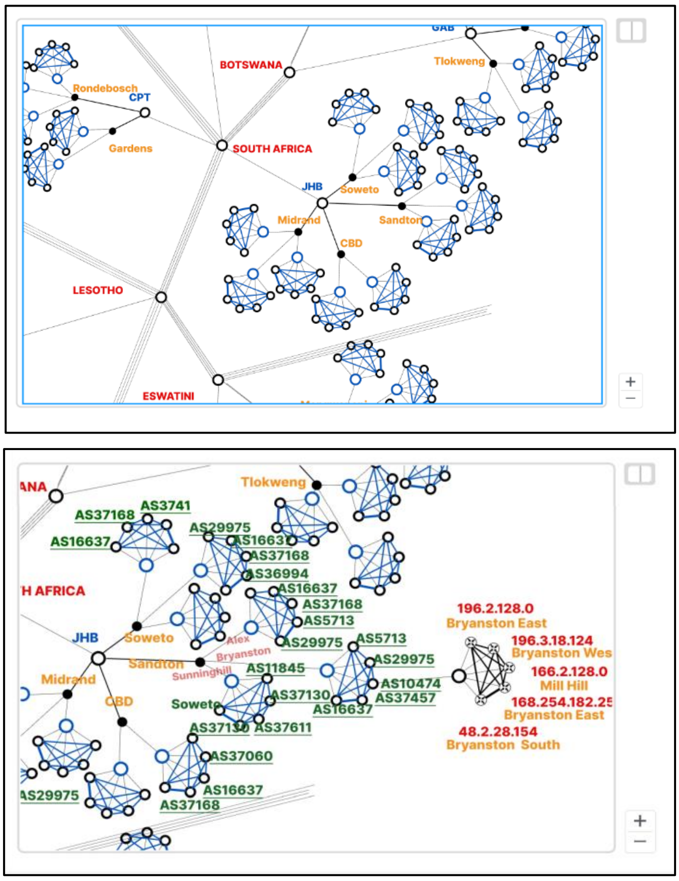


Figure 6 Tree Graph Based Prototype

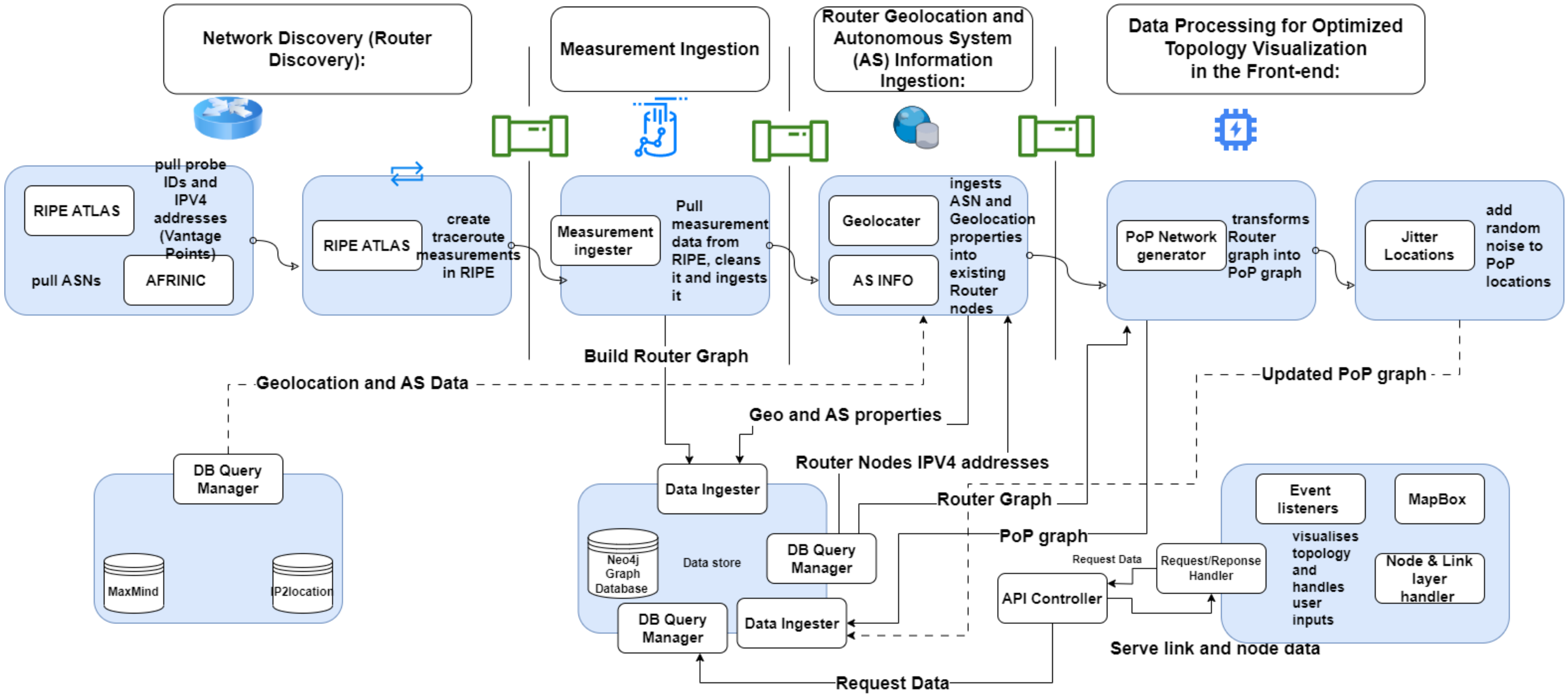


Figure 7 System Architecture

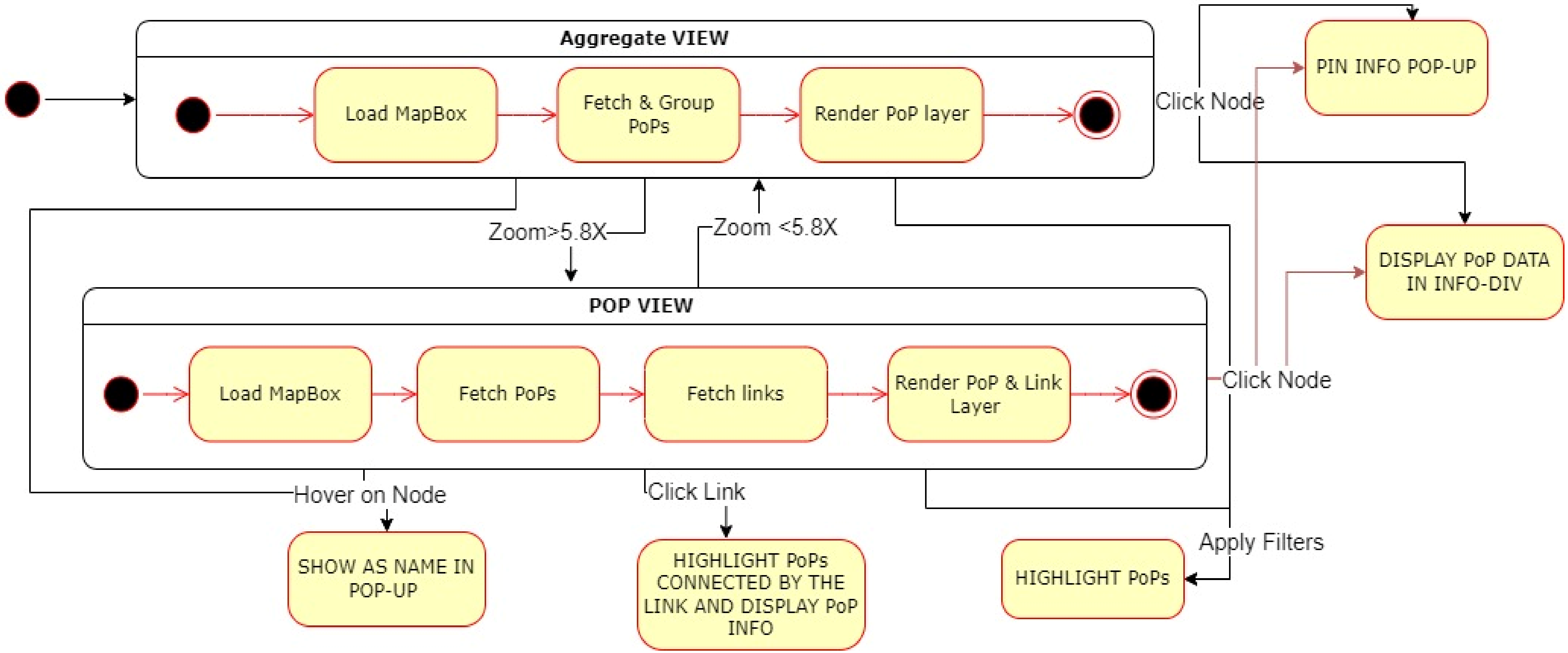


Figure 8 System States

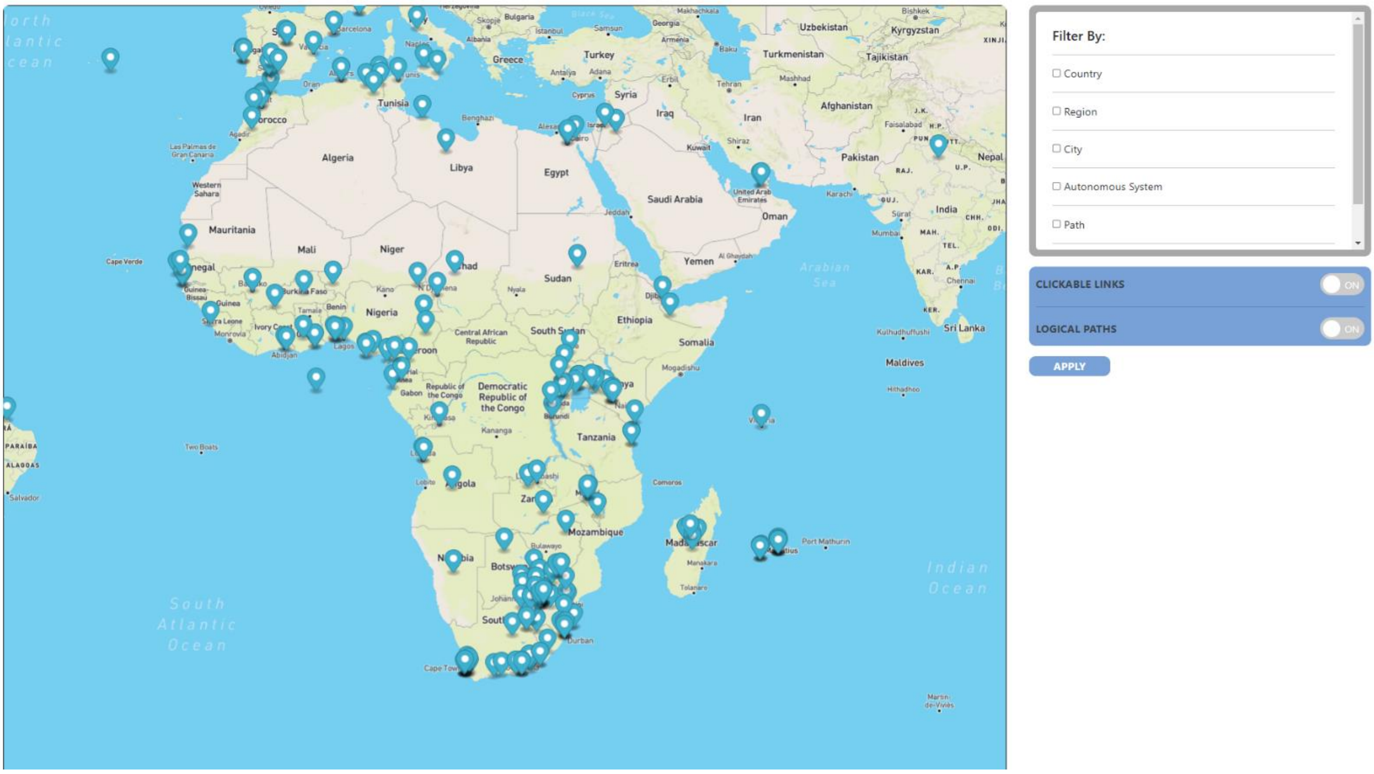


Figure 9 Final MAIT tool Landing Page

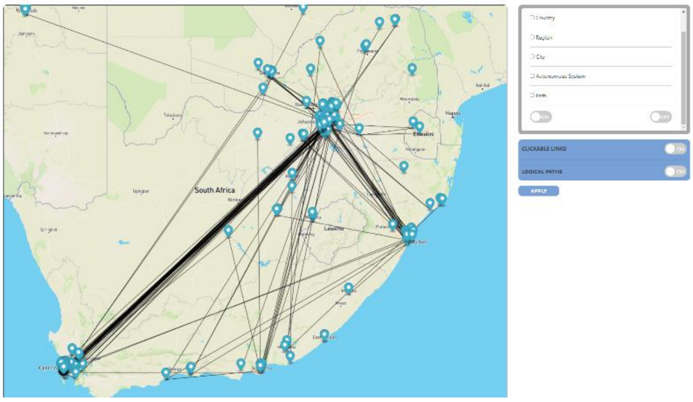


Figure 10 Interconnections Between routers visualised.

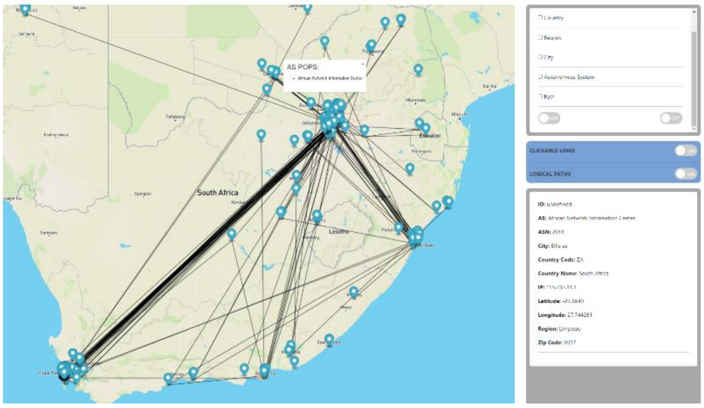


Figure 11 Router Marker Clicked and Information div displayed.

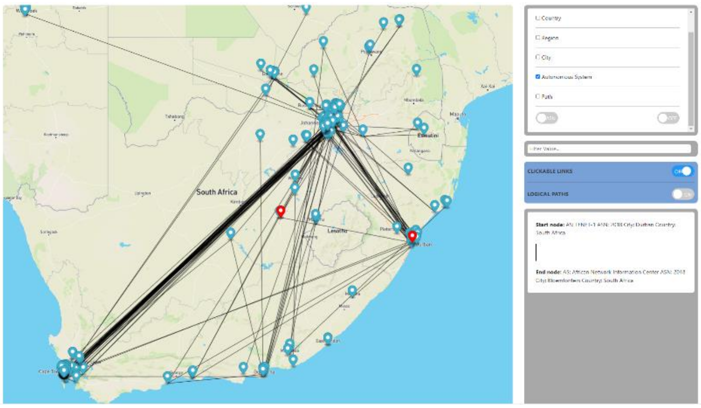


Figure 12 Interactions with links: link clicked.

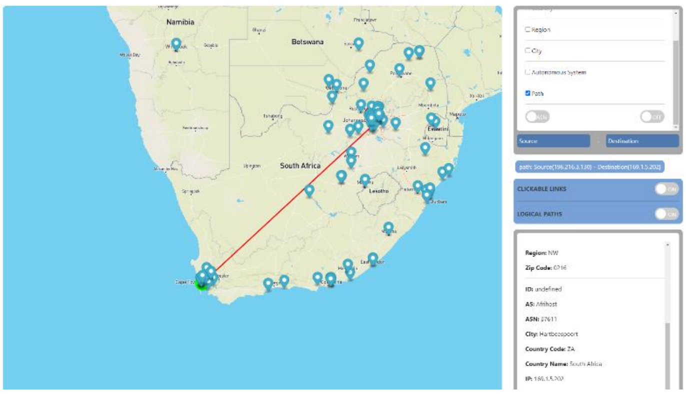


Figure 13 Shortest Path

Test Case	Inputs	Expected Behaviour	Results
Retrieving autonomous system Numbers from AFRINIC	FTP link to AFRINIC data	A list of Autonomous System numbers registered by AFRINIC written to 'afrinic_asns.txt' file	Passed
Pulling prefixes or address ranges assigned to an autonomous system via RIPE STAT	Autonomous System Number	A list of prefixes or IP ranges assigned to the AS packaged in a JSON	Passed
Retrieval of active probes using RIPE ATLAS API for probes	Country Code	List of IDs and IP addresses of the active probes in the specified country	Passed
Creating traceroute measurement in RIPE ATLAS	Probe ID and IP addresses of the target	Traceroute measurement running in RIPE ATLAS and the measurement ID is returned	Passed
Fetching measurement from RIPE ATLAS	Measurement ID	Measurement ID returned correlates with the measurement ID in RIPE ATLAS	Passed

Table 2 API Tests

Test Case ID	Integration Points	Input	Expected Output	Results
INT01	Integration of the creation, fetching and storage of measurements	Measurement creation and Data ingestion scripts	Processed data integrated seamlessly into Neo4j database, nodes and edges properly linked	Passed
INT02	System, GeoLite2/IP2Location to Neo4j Database	Known IP address with geolocation & AS details	Router node updated with geolocation and AS information	passed
INT03	Router Graph to RouterClone Graph	Cloning script	Creation of a 'RouterClone' graph identical to the original	passed
INT04	Preprocessing to Visualisation	Router focused graph	PoP focused graph for optimised for visualisation with reduced node count	Passed

Table 3 Integration Tests

Test Case ID	Component	Input	Expected Output	Results
TC01	Probe Data Extraction Script	RIPE Atlas API response for retrieving probe data	Successful extraction and storage of probe data	Passed
TC02	ASN Retrieval Script	AFRINIC API response for retrieving ASNs	Successful extraction and storage of ASNs	Passed
TC03	Prefix retrieval script	ASN Number	Successful extraction of IP Prefixes for given ASN	Passed
TC04	Zmap Scan Script	IP Prefix	Identification of reachable targets within the IP prefix range	Passed
TC05	Measurements script	Probe and Target IP Addresses; RIPE Atlas response once a measurement is created.	Successful creation of traceroute measurements and storage of Measurement IDs	Passed
TC06	Individual Traceroute Measurement Processing with the data ingestion script	Single traceroute measurement	Successful creation of router nodes and edges in the graph database.	Passed
TC07	MaxMind GeoLite2 Geolocation Lookup	Router IP	Successfully fetched geolocation details for a given IP and updated the router nodes with the data.	Passed
TC08	IP2Location Fallback Geolocation Lookup	Router IP	Fetch geolocation details for given IP if not found in MaxMind	Passed
TC09	Location jittering script	PoP coordinates in the same location	Slightly altered coordinates for overlapping PoPs, within 50km of original location	Passed

Table 4 Unit Tests