# Visualizing the Performance of an Anycast Network

Cas Wiefferink

**Supervisors** 

Dr. J. Zwiers W.B. de Vries

CREATIVE TECHNOLOGY 02-07-2019

UNIVERSITY OF TWENTE.

# Abstract

An interactive data visualization has been built that can visualize how an anycast network is functioning. Numerous interaction techniques have been utilized to allow the user to view, explore and analyze the data such that unusual routing patterns and deficiencies of the Border Gateway Protocol can be detected. The development of the project has been based on the human-centered design process making use of multiple iteration cycles. In each cycle a prototype was built and evaluated after which the received feedback was prioritized accordingly and taken in to account for the next iteration cycle. The final implementation supports drilling down in to the data such that the distribution of the queries can be viewed per continent, country, anycast site and autonomous system.

# Acknowledgements

First of all I would like to thank Dr. J. Zwiers, supervisor of this project, for his useful feedback and keeping me on track throughout this project. Secondly, this work would not have been possible without W. de Vries, critical observer and stakeholder, who has provided a huge help with guiding me through the design, implementation and evaluation phases of this project by giving constructive feedback and direction.

Finally, I would like to thank the people who were present during the SAND meeting for giving me the time to demonstrate my results and providing me with valuable feedback.

# Table of contents

	Abstract			
	Acknowledgements			
	LIST OF F	GURESV		
1	INTE	ODUCTION1		
	1.1	Problem		
	1.2	INTERACTIVE DATA VISUALIZATION		
	1.3	RESEARCH QUESTIONS		
	1.4	REPORT OUTLINE		
2	STAT	TE OF THE ART ON INTERACTIVE DATA VISUALIZATION		
	2.1	BACKGROUND INFORMATION		
	2.2	LITERATURE REVIEW		
	2.3	COMPARISON OF VISUALIZATION TOOLS		
	2.4	RELATED WORK		
	2.5	CONCLUSION		
2		TION 14		
3	IDLA	14		
	3.1	Метнод		
	3.2	IPACT ANALYSIS		
	3.3	USE CASE SCENARIO		
	3.4 Results			
	3.5 CONCLUSION			
4	SPEC	2IFICATION		
	4.1	Метнод		
	4.2	RESULTS		
	4.3	CONCLUSION		
5	IMP	LEMENTATION		
	5.1	Метнор		
	5.2	RESULTS		
	5.3	CONCLUSION		
6	EVA	LUATION		

6.1		SAND PROJECT	0		
6.2		RESULTS	0		
6.3		FINAL ITERATION	1		
	6.4	CONCLUSION	2		
7 CONCLUSION			3		
8 DISCUSSION			4		
9 FUTURE WORK					
A	APPENDIX				
	A1 – PEN-AND-PAPER PROTOTYPES				
	A2 – JAVA PROGRAM				
A3 – R Script					
R	REFERENCES				

# List of figures

FIGURE 1 – MAGIC QUADRANT FOR BUSINESS INTELLIGENCE AND ANALYTICS PLATFORMS (SALLAM ET AL., 2017)	8
FIGURE 2 – VISUALIZATION OF VERFPLOETER DATA (DE VRIES ET AL., 2017)	11
FIGURE 3 – VISUALIZATION OF RIPE ATLAS DATA, EACH CIRCLE REPRESENTING THE LOCATION OF AN ACTIVE VP (ABEN ET AL., 20	)17)12
FIGURE 4 – VISUALIZATION OF RIPE ATLAS DATA, ZOOMED IN ON THE NETHERLANDS	13
FIGURE 5 – THE HUMAN-CENTERED DESIGN PROCESS (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2010)	15
FIGURE 6 – THE INPUT DATA AFTER REMOVING UNWANTED ATTRIBUTES FROM THE RAW DATASET	21
FIGURE 7 – OVERVIEW OF THE DIFFERENT TOOLS USED FOR THE PROJECT	22
FIGURE 8 – CLASS DIAGRAM OF THE INPUT DATA FOR THE VISUALIZATION (OUTPUT OF THE R SCRIPT)	22
FIGURE 9 – CLASS DIAGRAM OF THE OUTPUT OF THE JAVA PROGRAM BEING COMBINED WITH OTHER LOCATIONAL DATA	22
Figure 10 – Screenshot of prototype 1	24
Figure 11 – Screenshot of prototype 2	26
Figure 12 – Screenshot of prototype 3	27
Figure 13 – Screenshot of prototype 4	
Figure 14 – Screenshot of prototype 5	29
FIGURE 15 – SCREENSHOT OF THE FINAL PROTOTYPE	32

# **1** Introduction

## **1.1 Problem**

The Border Gateway Protocol (BGP) is responsible for selecting the optimal anycast site for a given user, which often is the one closest to the user, although it does not always succeed in doing so. This can cause a user's request to be delivered to an anycast site that is not optimal, increasing the latency that is experienced and increasing the load on the network. Detecting these outliers proves to be a difficult task.

Recently de Vries et al. (2017) have developed Verfploeter, which is a powerful tool that can collect large amounts of data about the catchments of an anycast network. However, they have visualized the collected data in a static way and the visualization is therefore quite limited in use, which will be further discussed in paragraph 2.4.1. In this research project an interactive data visualization has been developed to make it easier to view, explore and analyze the data that has been gathered using Verfploeter.

Simply visualizing the data does not necessarily mean you can extract useful information from it and certainly not knowledge (Rowley, 2007). The aim is to show, however, that the data can be a source of valuable information that can be used to improve the anycast network. For example, the data can be used to detect whether there are Vantage Points (VPs) that query anycast sites that are not geographically closest to them. This could be the result of a misconfiguration of the BGP or have other causes as discussed in paragraph 2.1.3. Once this is known and has been verified, something can be done about it such that it can be prevented in the future.

# **1.2 Interactive data visualization**

An interactive data visualization implemented as a dashboard could be a solution to this problem and to overcome the limitations of a single, static visualization. The dashboard can be used to perform exploration and analysis of the data in a visual way. Seen as in visual analytics, which is a research field where interactive visualizations are used for understanding, reasoning and decision making on the basis of a large dataset (Kohlhammer, Keim, Pohl, Santucci, & Andrienko, 2011). It can be used to explore and analyze the data, in the sense of detecting the expected and discovering the unexpected. For example, to detect the outliers in the dataset, which users query an anycast site that is not geographically closest. Or to discover unusual anycast routing patterns and deficiencies, such that they can be acted upon.

# **1.3 Research questions**

In this research project, a visualization implemented as a dashboard for researchers and network operators is proposed that allow them to view, explore and analyze information about the anycast network. The main research question and sub questions can then be defined as:

- "How to implement an interactive data visualization that can visualize the functioning of an anycast network?"
  - "How to enable the viewing, exploration and analysis of data using an interactive data visualization?"
  - "What are the design requirements and what functionalities must the visualization support?"

# 1.4 Report outline

This report is divided in to 6 main chapters. The second chapter discusses relevant background information for this research and different interaction techniques that can be implemented in a visualization. Furthermore, it includes a comparison of the tools available for visualizing and will be looked at related work to this project. Chapter 3 describes the ideation phase and focuses on gaining a better understanding of how the interactive data visualization should be designed. In chapter 4 the design ideas are translated in to a prioritized list of system requirements. Chapter 5 starts with how the data has been organized and prepared for the visualization and it also describes the implementation process. In chapter 6 the prototype is evaluated and the final version is presented. The final chapters conclude this project. It discusses the research questions and gives recommendations for future work.

# 2 State of the art on interactive data visualization

This chapter will focus on the state of the art of the different aspects within this research project. Starting with some background information and the different concepts relevant to this research. It includes a literature review about the differences and similarities between interaction techniques that can be applied to make an interactive dashboard. Next, a comparison is made of the tools that are available for building a data visualization. Furthermore, it discusses related work including a visualization of data gathered by Verfploeter and an interactive data visualization closely related to the topic.

## 2.1 Background information

In this section different concepts relevant to this research will be briefly discussed to provide some general background information. For more in-depth explanations there are references to other sources.

### 2.1.1 Anycast

The Internet recognizes five main addressing and routing methods. The most popular amongst them is unicast, where there is one sender and one destination. A lesser known, growing in popularity amongst network operators (Abley & Lindqvist, 2006), method is anycast. Pillai (2014) explains well what anycast is and how it works. Basically, anycast delivers the message of one sender to any one out of a group of destinations, typically to the one that is nearest. This particular addressing and routing method has its main use in the Domain Name System (DNS), but is also quite common in Content Delivery Networks (CDNs).

With anycast there are multiple destinations a request from a user can be delivered to, these destinations will be called anycast sites<sup>1</sup> in this report. All anycast sites that provide the same service are assigned to a single Internet Protocol address (IP address), also called the anycast address. A user wanting to make use of this service would send a request to this anycast address, the request will be delivered to the site nearest to the user. Determining which anycast site is nearest is handled by the Border Gateway Protocol (BGP). This protocol takes care of dividing the Internet in to regions called catchments, each served by a different anycast site.

### 2.1.2 Mapping catchments

Mapping anycast catchments is done using Vantage Points (VPs). VPs are devices or computers that can be used to measure which anycast site it queries from the location it is in. There are commercial tools

<sup>&</sup>lt;sup>1</sup> An anycast site can consist of 1 or multiple servers at a certain location

available such as RIPE Atlas by RIPE NCC Staff (2015) that can map anycast catchments using thousands of VPs. RIPE Atlas makes use of active VPs, dedicated hardware devices spread out over the world to conduct their measurements. Their coverage is however inconsistent. Recently de Vries et al. (2017) have developed Verfploeter, which is a new approach to map anycast catchments covering the vast majority of networks around the globe. Verfploeter uses 430 times more VPs than the approximately 9000 VPs RIPE Atlas makes use of, providing a better coverage. Instead of using active VPs, Verfploeter uses passive VPs which can be any computer connected to the Internet that replies to queries asking for a response. This allows for the spotting of routing differences, especially in the regions that have sparse coverage from RIPE Atlas, like South America and China.

### 2.1.3 Malicious incentives

Paragraph 1.1 discussed that vantage points could query an anycast site that is not geographically closest to them due to a configuration mistake. However, this may not always happen by accident, as it could also be done on purpose with a malicious intent. This is known as BGP hijacking, where Internet traffic can be misdirected, monitored, intercepted, black holed or directed to fake websites as part of a man-in-the-middle attack (Cloudflare, n.d.). An example of this was in 2016, where the Chinese Internet Service Provider (ISP) China Telecom has been misdirecting large quantities of Internet traffic via China before delivering it with a small delay (Demchak & Shavitt, 2018). The researchers were able to identify patterns that suggested deliberate hijacking and monitoring of the traffic to possibly aid them in the surveillance of Western countries and companies. For this reason, it is a rather politically sensitive subject, especially with world powers involved such as China and the United States, but also countries like Russia and Pakistan (Cloudflare, n.d.).

#### 2.2 Literature review

Information overload is a familiar phenomenon of the information age. Due to the ever increasing of computer power and storage capacity over the last decades, data is produced and collected at an astonishing rate. The data is collected because people believe that it is a potential source of valuable information, which may provide a competitive advantage in the future. Finding the valuable information hidden in the data however, is a challenging task (Keim, 2002; Keim, Mansmann, Schneidewind, & Ziegler, 2006).

Thus, methods must be explored on how to interpret the huge amount of data in order to understand and extract knowledge from it. Giving a user the ability to interact with a data visualization proves to be helpful in different ways. Kosara, Hauser and Gresh (2003) point out that interaction can help the user to better understand the data that is visualized or even understand it at all. Besides it can be utilized to realize analytic goals users may have. Therefore, the objective of this section is to understand the similarities and differences between the techniques that can be used to make a data visualization interactive.

This section covers multiple, low-level interaction techniques, including filtering, selecting, brushing and linking, zooming and details-on-demand. To gain a better understanding of the similarities and differences, for each technique definitions from multiple authors will be reviewed and evaluated. Covering all the techniques that can be used for interaction is beyond the scope of this review, it will limit to arguably the most common techniques used.

#### 2.2.1 Filtering

There is no clear definition of filtering within the context of interactive data visualization. Yi, Kang, Stasko and Jacko (2007) describe filtering as changing the set of data that is presented based on specific conditions set by the user. When the data falls outside the range, it is not shown. This goes also by the name of querying, as suggested by Keim (2002). Besides querying, Keim defines the technique browsing as something that "can be done by a direct selection of the desired subset" (p. 5) instead of filtering indirectly. This complies with Khan and Khan (2011) who describe filtering as any means to decrease or increase the information quantity that is displayed. Within this report, filtering will be defined as an overarching term that includes both the indirect filtering of data, also called querying, and the direct filtering of data, browsing.

#### 2.2.2 Selecting

Closely related to filtering, is the technique selecting, which is sometimes used ambiguously amongst researchers of interaction techniques. Yi et al. (2007) describe selecting as the ability to mark data points of interest to keep track of it. In contrast to filtering the rest of the data is not hidden, instead it makes the selection visually distinctive. Khan and Khan (2011) stand by this definition of selecting. Furthermore, they observe that the two techniques that fall under filtering, direct and indirect filtering, can also be applied to specify a selection. As mentioned earlier, Keim (2002) uses the phrase "direct selection" to describe a sub technique of filtering, which may be confusing. Within this report, selecting will be defined as an overarching term that includes both the indirect selection of data and direct selection of data.

#### 2.2.3 Brushing and linking

A definition of linking cannot be given without covering brushing, as these techniques are often used in combination with each other. Doleisch (2004) demonstrates that brushing can be defined as marking a subset of the data as being of special interest. Interestingly, this is the same definition that was given to selecting, but when selecting is used in combination with linking, then selecting is called brushing. Highlighting the brushed subset in multiple visualizations such that all visualizations highlight the same data points at the same time, is called linking (Wilkinson, 2005). Keim (2002) adds to this that linking two or more visualizations to each other should make it possible to detect correlations and dependencies between them. Yi et al. (2007) also call the combination of these techniques connecting, whereas Shneiderman (1996) names it relating. The definition of Wilkinson (2005) will be used within this report, where linking and brushing is defined as highlighting a brushed subset in multiple visualizations at the same time.

#### 2.2.4 Zooming

The ability to zoom in and out as an interaction technique appears simple on the surface, but authors seem to partially disagree on its definition. Craft and Cairns (2005) give a general definition of zooming where it "refers to the adjustment by the user of the size and position of data elements on the screen" (p. 2). To be more precise, zooming in enlarges smaller data points and zooming out produces the opposite result by reducing larger data points without altering its representation (Figueiras, 2015). Yet, Keim (2002) claims that zooming in does not only mean to display data points larger but it should also change the data representation to present more details or attributes per data point. Khan and Khan (2011) maintain this concept of zooming. Despite the controversy, in this report the term zooming will be used as merely adjusting the size and position of data elements.

#### 2.2.5 Details-on-demand

The final interaction technique to be reviewed, which partially overlaps in definition with zooming, is to view additional details on demand. Two ways to explain details on demand have been identified. Craft and Cairns (2005) and Dix and Ellis (1998) highlight that providing in-depth details about all the displayed data elements in a data visualization is impractical. This is mainly caused by the limitations of the screen size and the visualization becoming too complex to understand. The first to suggest the details-on-demand technique was Shneiderman (1996), who outlined it as selecting one or multiple data elements and get extra details when needed. Interestingly, Shneidermans definition of details-on-demand is seen as a component of zooming by Keim (2002) and Khan and Khan (2011) as was seen in

the previous paragraph.

The second way to describe details on demand is by distinguishing two different ways to implement it. Figueiras (2015) observes two different sub techniques of details-on-demand, either by providing drill-down options or by the use of tool-tips or pop-ups. Whereas the use of pop-ups is quite self-explanatory, drilling down is not. Drilling down, as defined by Jensen, Pedersen and Thomsen (2010), offers more detail by expanding one dimension of data in to multiple dimensions, going down a level in the hierarchy. Within this report, details-on-demand will be defined as an all-encompassing term that includes both drilling-down and tool-tips.

#### 2.2.6 Conclusion

The aim of this literature review was to understand the similarities and differences between arguably the most common interaction techniques used in data visualization. To answer this question, a literature study was executed. For each technique a clear definition was found and in the process of doing so, a better understanding of the similarities and differences between the techniques was achieved.

A fair amount of inconsistencies between definitions of different authors were found. To begin with, the term 'selecting' can be easily misinterpreted as its either used as a name of an interaction technique or its used to describe a sub technique of filtering. Not only can this be used to describe a sub technique of filtering, it can also be used to describe a sub technique of selecting. When selecting is used in combination with linking, selecting is known as brushing. The combination of brushing and linking is also referred to as connecting and relating. Finally, the definition of zooming is disagreed upon amongst authors. Where some defined zooming as merely adjusting the size and position of data elements, others argued it should also include presenting extra details or attributes per data element.

Unquestionably, there is room for further research. As was stated in the introduction, this review describes the most common interaction techniques. Yet, the lesser known interaction techniques may also be very interesting to implement in a data visualization. These could be researched in a similar fashion to this literature review. This literature review gives an overview of how interaction techniques can be defined, but not a single source was found that extensively described advantages and disadvantages. Knowing the answer to what the advantages and disadvantages of a particular interaction technique are, would be extremely helpful with implementing the technique effectively. One of the reasons there is little known about this, is the complexity produced by all of the different interaction techniques that can be applied. To get answers to these kinds of questions, user tests should be performed.

## 2.3 Comparison of visualization tools

Currently there are three market leaders in the area of business intelligence: Tableau, Qlik and Microsoft (Sallam et al., 2017), see Figure 1. These will be compared with each other based on features that are demanded for this project. Although it is also possible to build a visualization using other tools, these frameworks offer a solid foundation to build upon.

Other commonly used tools for data visualization in this discipline are open source JavaScript libraries such as D3 and Leaflet. D3 is a graphing library using HTML, SVG and CSS and offers basic building blocks like bars, nodes and edges. Leaflet is a mapping library, including geographic displays, markers and grouping. The benefit of using these tools is that they allow for highly customizable visualizations. However, the drawback is that they are relatively low-level in comparison to the previously mentioned tools. As a consequence, using D3 or Leaflet would require significantly more time to be invested before a useable prototype can be presented. This is undesirable as it would not allow for multiple iteration cycles which is key for building prototypes. Therefore, they have been left out of the comparison.



Figure 1 – Magic Quadrant for Business Intelligence and Analytics Platforms (Sallam et al., 2017)

The tools will be compared based on the demands for this project. The demands that have been identified are:

*Visualization types* – The amount of visualization types that are supported gives a good insight in to how flexible the tool is.

*Maps* – This project deals with geospatial data, therefore the ability to plot this data onto a map may be useful, as can be seen in paragraph 2.4.

*Interactive dashboard* – A dashboard allows the user to view and interact with multiple visualizations at the same time. As known from the literature review, this helps detecting correlations and dependencies in the data.

*R* integration – R is a programming language for statistical computing and graphics. R integration within the tool may be useful, as it can be used to build highly custom data visualizations.

*Interaction support* – As was found in the literature review, interaction with a data visualization plays an important role in the exploring and analyzing of data.

*Technical support* – The amount and quality of technical support that is available for the tool. The documentation, size of the user base and availability of tutorials of a tool play an important role in the technical support.

*Publish to website* – Whether the tool has support for publishing the data visualization to a website. As of now, it is not certain whether this feature will be useful, but it is important to take in consideration.

	Tableau	Qlik	Microsoft Power BI
Visualization types	24	15	24
Maps	Available	Not available, only after purchase	Available
Interactive dashboard	Possible	Possible	Possible
R integration	Yes	Yes	Yes
Interaction support	Decent	Moderate	Good
Technical support	Moderate	Moderate	Good
Publish to website	Not available, license required	Possible	Possible

#### Table 1 - Comparison of the three tools, adapted from van Bruxvoort (2018)

Based on the comparison in Table 1, Qlik offers the least of functionalities out of the 3 tools. It supports only 15 different visualization types, no maps without purchasing the product and a moderate interaction support. Tableau and Microsoft Power BI on the other hand are very comparable in respect to the features they offer and thus are good contestants of each other as they both are likely to meet the requirements for this project.

Tableau offers slightly less interaction support and it does not allow you to publish to a website without buying a license. Although, financial matters should not be a motive for choosing one tool above the other, it does play a minor role when the different tools are that similar to each other. Furthermore, Microsoft Power BI seems to have the most active community, making the technical support excellent in contrast to the other two tools. Therefore the choice has been made to go forward with Microsoft Power BI. If for some reason Microsoft's framework appears to be not suitable for the project, Tableau remains a good second choice and will thus serve as a back-up.

### 2.4 Related work

#### 2.4.1 Data visualization of Verfploeter

The first data visualization that will be discussed is from de Vries et al. (2017) depicted in Figure 2. It visualizes the catchments of 8 different anycast sites. For each block of Vantage Points a pie chart is drawn in the country of origin, where its size indicates the amount of VPs it contains. The color tells how the queries are distributed between the anycast sites, where each site has its own distinctive color and a three-letter airport code from which the location of the site can be deduced.

The world map gives an overview of how the blocks are mapped across the globe and where most of queries end up from a given block. For example, knowing that CPH stands for Copenhagen and ENS stands for Enschede, you can spot that most of VPs in Europe query a site that is within Europe. Nevertheless, the visualization has several limitations. One of the disadvantages of using pie charts is that it is relatively hard to compare the sizes of the slices within a pie chart but also between different pie charts (von Huhn, 1927). Furthermore, the areas on the map where multiple, large blocks are located the pie charts overlap with each other and it becomes cluttered. Besides, the colors of the pie charts of the small sized blocks are indistinguishable, making it impossible to see the color distribution of the pie charts. While the visualization is adequate to give an overview of the data, it is hard to draw any meaningful conclusions from this visualization.



Figure 2 – Visualization of Verfploeter data (de Vries et al., 2017)

#### 2.4.2 Interactive data visualization of RIPE Atlas

The next data visualization that will be discussed is an interactive one by RIPE Atlas (Aben, Buddhdev, & Petrie, 2017). As mentioned earlier in paragraph 2.1.2, RIPE Atlas is a commercial tool that maps anycast catchments using active VPs. They have also made an interactive data visualization using the data they collected of which a screenshot can be seen in Figure 3. Each circle represents the location of a VP and its color is determined by the physical location of the DNS server it queries during measurements. Colors indicate the continent: Africa (brown), Europe (blue), South America (orange), Asia (red), North America (green), Middle East (yellow) and Oceania (pink). The map supports different kinds of interaction. You can zoom in on a specific area of interest and it can filter on location of the anycast site, showing only VPs that query certain anycast sites or certain continents. Furthermore, it is possible to choose the time of measurement.

The world map gives an indication of how the VPs are spread around the globe. When zooming in the circles scale automatically, this helps maintaining a good overview, see Figure 4. Yet, it also comes with deficiencies. When fully zoomed out the map feels fairly cluttered in the areas that are densely packed with circles. A similar problem the previously discussed data visualization also faced. This time however you are able to filter out the undesired data making it somewhat clearer. The data visualization feels sluggish when loading and after applying filters. Initial loading of the visualization takes about 30 seconds and loading after applying a filter takes roughly 10 seconds. They chose to use the color of the circle's outline to indicate a specific DNS server within a continent. Although this is quite clever use of color in theory, it is implemented in such a way that the outlines are barely visible and thus hard to distinguish from the other outline colors. Furthermore, the visualization is not focused on the detection of outliers, but more on giving an overview. While it is possible to spot potential outliers here and there,

the volume of data is not sufficient to identify whether there is more to it than just those one or two outliers. It does not give any indication of the scale of potential outliers. Finally, the ability to explore and analyze is minimal as the visualization is limited to a single representation of the data, a world map.



Figure 3 – Visualization of RIPE Atlas data, each circle representing the location of an active VP (Aben et al., 2017)



Figure 4 – Visualization of RIPE Atlas data, zoomed in on the Netherlands

### 2.5 Conclusion

In this chapter the state of the art on data visualization with respect to the anycast protocol has been examined. This has been split up in to four parts. The first part provided some background information and the different concepts relevant to this research. The second part covered multiple interaction techniques that can be applied to a data visualization by means of a literature review. For each technique a clear definition was found and a better understanding of the similarities and differences between the techniques was achieved. In the third part different visualization tools have been compared based on the features which are thought to be important. From this comparison can be concluded that Microsoft Power BI seems to be the best fit for this project. In the final part two data visualizations have been reviewed. Both of them contain relevant elements that could be incorporated in to an interactive data visualization of an anycast network. But also several areas of improvement were found, which should be kept in mind so that they can be overcome.

This chapter shows that research in to the development of an interactive data visualization focused on exploring and analyzing data of an anycast network and the detection of outliers is novel, since there is no such application available as of now.

# **3 Ideation**

The reason behind having an ideation phase is to gain a better understanding of how the interactive data visualization should be designed. Ideation is an important phase as it broadly determines the project's specifications. As was already mentioned in chapter 1, the main focus of the data visualization will be on exploring and analyzing the data. This is however very broad and should be narrowed down to an elaborated idea, which will be the focus of this chapter. This will be achieved while being led by the research questions set in paragraph 1.3. Furthermore, this chapter focuses on understanding and specifying how the data visualization could be used in practice through an iPACT analysis and a use case scenario.

# 3.1 Method

In order to get to an elaborated idea different techniques and methods will be used. The development during the project will be based on the human-centered design process, as can be seen in Figure 5. An important aspect using this process is an iterative design approach. Low-fidelity prototypes and mock-ups of potential designs help to obtain a deeper understanding of the user requirements. Thus feedback can be obtained on early design concepts before the requirements are finalized.





# 3.2 iPACT Analysis

Before the user requirements can be specified it is important to understand and specify the context in which the system will be used. By means of an iPACT analysis will be looked at the envisioned system through the eyes of the user. This method is used to describe the system's *intention*, the *People* who will use this system, the *Activities* that are related to the system, the *Context* in which the system would be used and finally the *Technology* that is used to realize the envisioned system.

#### 3.2.1 Intention

The intention of the envisioned system, the interactive data visualization, is to give users a tool to perform visual analytics. As was already briefly touched upon in paragraph 1.2, visual analytics is a research field that focuses on using interactive visualizations to understand, reason and make decisions on the basis of a large dataset (Kohlhammer et al., 2011). Thus the main goal would be not only to represent the data visually, but also giving users the ability to explore and analyze it through interaction such that misconfigurations in the BGP can be detected.

#### 3.2.2 People

The people that this project is aimed for are network operators and researchers in this area of expertise. Therefore can be assumed that they have a technical background and know how to handle basic computer programs. It is unlikely that any potential user of this system is computer illiterate. This must not take away that the system should be user friendly, but it can be designed whilst keeping this in mind.

#### 3.2.3 Activities

Three main activities can be identified in respect to this system: viewing, exploring and analyzing. With viewing is meant that the interactive data visualization allows the user to gain an overview of how the anycast network is performing. In the literature review in paragraph 2.2.5 was found that providing indepth details about all the displayed data elements in a data visualization is impractical. For this reason details-on-demand should be used if it is desired to further explore the data. Exploring of data can be achieved by offering more detail by expanding one dimension of data in to multiple dimensions, drilling-down a level in the hierarchy. For example, instead of viewing the division of queries per continent, you can view the division per country or autonomous system. Finally the system can be used to analyze, as in detecting whether the visualization shows what is expected but also discovering the things that may be unexpected.

#### 3.2.4 Context

These activities occur in a certain context. There are different reasons to use the system to analyze the anycast network, such as to discover unexpected or unusual traffic between countries. It could for example happen that a significant portion of the traffic from the United States is routed through or to China, whereas it is expected for the traffic to stay within the borders of the US. This could be the result of some accidental misconfiguration, but may also be done on purpose for illegitimate motives, such as to take down a website or even espionage. The system can also be used for the opposite reason, to verify whether the traffic flows as expected. An example of this could be to ensure that most of the traffic from the Netherlands stays within the borders of Europe.

#### 3.2.5 Technology

Different technologies are involved in to building the data visualization, these include:

Java (programming language) – Although datasets are available to use, Java will be used to collect additional information complementing the data.

R (programming language) – R will be used to clean and restructure the data such that it can be used for the data visualization.

Microsoft Power BI (computer program) - Microsoft Power BI will be used as the foundation of the data

visualization.

*DAX* (Data Analysis Expressions) – DAX is a functional language that is used to program the logic behind the data visualization.

These technologies and how they are connected to one another will be further discussed in chapter 5.

### 3.3 Use case scenario

John works at an Internet Service Provider in the United States as a data engineer. His main job is to monitor the performance of their network. Although many of this has been automated where algorithms are used to detect potential issues, still many slip through and therefore remain unnoticed.

One day John suddenly receives a call from the customer support department. They are receiving a lot of complaints about sluggish Internet speeds in a suburb of Los Angeles and ask him whether he can look in to it. He agrees and starts scrolling through the error log, but there seems nothing out of the ordinary as there are no substantial errors given. Manual inspection is required, so he starts looking at the data visualization, which helps giving him a good overview of the situation.

When he zooms in on the US he notices something unusual. It appears that a significant percentage of the traffic leaves the country's borders. Whereas typically over 95 percent of the traffic remains within US borders, now this has dropped to 89 percent. When further drilling down in to the data it appears that 7 percent is routed through China. To be more specific, to two anycast sites located in China, one is located in Hong Kong and the other is located in Shenzhen, which explains the decreased Internet speeds. John wonders whether it's a result of some accidental misconfiguration or whether it was done on purpose and passes the case on to his manager.

### **3.4 Results**

After multiple brainstorming sessions early design sketches were made, which can be found in Appendix A1. In these sketches different design ideas were showcased to obtain feedback from the supervisors such that there can be iterated upon. The main ideas are based on the view of a world map in combination with some other type of graph, in this case a bar chart. Each country in the world map is assigned a gradient of color based on a data aspect of interest. One of these aspects could be for example the percentage of instances that end up querying an anycast site that is over a specified distance between the source and destination. The distance can be set using a slider or some other form of user input.

During meetings various ideas came to mind that may be interesting to implement in to the

visualization. First of all, instead of filling a country with a color (also known as a choropleth), it would be better to assign the color to a circle with a fixed size that is drawn for each country, a bubble chart. This helps preventing an unconscious bias, because countries with a larger surface area would seem more important than smaller countries even though this may not be the case at all. Furthermore, drawing conclusions using the distance between the user and anycast site can be unreliable. Although the exact location of the anycast sites are known, the user's location is only accurate on a country basis. For instance, what would be the distance between Germany and Enschede? That is highly dependent on the exact location in Germany. Finally, a more appealing idea would be to visualize the percentage of queries that end up at an anycast site in a foreign country. This, on the other hand, can be deduced with certainty from the dataset.

# 3.5 Conclusion

The goal of this chapter was to gain a better understanding of how the interactive data visualization should be designed. Before this could be achieved, first another step in the human-centered design process had to be overcome, which was specifying the context in which the final product will be used. This was done by means of an iPACT analysis describing the system's intention, People, Activities, Context and Technology. After this, different design ideas and concepts where discussed that were the result of brainstorming and feedback sessions with supervisors while making use of low-fidelity prototypes. In the chapters 4 and 5 these ideas and concepts will be further specified and realized respectively.

# **4** Specification

In chapter 3, the ideation phase, a better understanding was gained on how the interactive data visualization should be designed. The ideas and concepts that were found in that chapter need to be translated in to system requirements, which will be the focus of this chapter. The idea behind listing all the requirements is that they can be implemented in a structured way in the next phase of the project, the implementation phase.

# 4.1 Method

Prioritizing the system requirements will be done using the MoSCoW method, firstly introduced by Clegg and Barker (1994). The term MoSCoW is an acronym derived from the first letter of each prioritization category in which a system requirement can fall: *Must have, Should have, Could have* and *Won't have*. Where the Must haves are critical for the project being a success, the Should haves and Could haves are respectively important and desirable, but not necessary. The system requirements that fall in the latter two categories will be implemented if time and resources permit so. The Won't haves are either requirements that have been dropped or can be considered for future work.

# 4.2 Results

The system requirements sorted using the MoSCoW method can be found below in Table 2.

The visualization must	Discussed in paragraph		
Allow the user to view, explore and analyze the data	1.1		
Be able to zoom in on an area of interest	5.2.2		
Make use of different interaction techniques, implemented in a meaningful manner	2.2		
Support drill-down features to see the distribution per continent, country, anycast site and autonomous system	3.2.3		
The visualization should			
Not be biased to countries with a relatively large surface area	3.4 and 5.2.1		
Make use of relative numbers, but absolute values should also be accessible	5.2.2		
Be able to filter countries in an intelligent way, such that only the data	5.2.3		

Table 2 – System	n requirements	sorted using	the MoSCoW	method
------------------	----------------	--------------	------------	--------

remains that is most likely to be of interest to the user		
The visualization could		
Be accessible through a web page	2.3 and 5.2.5	
Have a search functionality to only show a specific continent, country, anycast	5.2.5	
site or autonomous system		
The visualization won't		
Visualize temporal data (data over time)	9	
Support hot swappable datasets	9	
Support datasets with other data structures	9	

# 4.3 Conclusion

The goal of this chapter was to translate the ideas and concepts, that were found in the previous chapters, in to a prioritized list of system requirements. For each requirement is mentioned in which paragraph it was or will be discussed. Not all requirements listed above were found in the ideation phase. Several were added during the iteration process as will be further discussed in paragraph 5.1.2. The prioritized list of system requirements can be used to build the visualization in a structured way in the next phase of the project, the implementation phase.

# **5** Implementation

In the previous chapter the design ideas and concepts were translated in to a list of system requirements divided in to four categories with different levels of importance. The focus of this chapter will be using those results to implement the visualization in a structured fashion. Before discussing the implementation process leading to the end result, first will be reviewed what techniques were used in the process and how they are connected with one another.

# 5.1 Method

### 5.1.1 Data organization

One of the most important and time consuming aspects of building a data visualization is organizing the data. As was briefly touched upon in paragraph 3.2.5, two programming languages have been used to prepare the data for the visualization, Java and R. The dataset shared by de Vries was in the format of a CSV file (Comma Separated Values). This is also the format that has been used throughout the data preparation phase.

The raw dataset initially contained several attributes that will not be of any use for the visualization and therefore have been filtered out. A class diagram of the filtered, raw dataset can be seen in Figure 6 and contains two attributes: The IP Address and the International Air Transport Association (IATA) airport code of the anycast site. The filtered, raw data contains over 4 million measurements/entries, where



Figure 6 – The input data after removing unwanted attributes from the raw dataset

each entry contains the IP Address of a computer and to which anycast site its query would be delivered to.

The IP Address can be used to find out the approximate geological location of the computer, i.e. its country, and to which autonomous system (AS) this computer belongs. In theory it's also possible to determine its city, but this has not been used due to a high rate of inaccuracy. Retrieving the country and autonomous system has been achieved using Maxmind's GeoLite2 databases along with their GeoIP2 API for Java<sup>2</sup>. The Java program written for this purpose, which can be found in Appendix A2, is the first step in the data organization process of which an overview is given in Figure 7.

<sup>&</sup>lt;sup>2</sup> Maxmind databases and API - https://dev.maxmind.com/geoip/geoip2/geolite2/



#### Figure 7 – Overview of the different tools used for the project

The resulting dataset of the initial processing and cleaning done by the Java program can be seen in the bottom left of the class diagram in Figure 9. Now the IP addresses in the dataset have been matched with their country and AS, however this is not yet sufficient. Since the goal is to calculate percentages, such as the percentage that ends up querying an anycast site in a foreign country or another continent, more information is required.





Figure 9 – Class diagram of the output of the Java program being combined with other locational data



Figure 8 – Class diagram of the input data for the visualization (output of the R script)

The third and final step is to construct the data visualization itself, which will be discussed in the following paragraphs.

#### 5.1.2 Iteration cycles

Part of the human-centered design process, as explained in paragraph 3.1, is making use of so-called iteration cycles. In each iteration cycle a prototype is built, either built from the ground up or improved upon a previous iteration. When the prototype is complete, it is evaluated with a stakeholder and

<sup>&</sup>lt;sup>3</sup> https://community.periscopedata.com/t/63fy7m/country-centroids and https://datahub.io/core/airport-codes

<sup>&</sup>lt;sup>4</sup> https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html

<sup>&</sup>lt;sup>5</sup> https://cran.r-project.org/web/packages/magrittr/vignettes/magrittr.html

supervisor of the project, i.e. with de Vries. The feedback that is received after each evaluation session is prioritized accordingly and will be taken in to account for the next iteration cycle. If the feedback contains something essential to the project that should be changed or added as a feature then the system requirements are also changed to align with the received feedback. Hence, the initial requirements that were set in the previous chapter were not set in stone. On the contrary, they keep evolving in to the final specifications until the final iteration cycle is concluded.

### 5.2 Results

In this section each iteration cycle is briefly discussed. This includes a description of the main things that changed in respect to the previous iteration, the feedback that was received and a screenshot of the prototype.

### 5.2.1 Iteration 1

The first prototype is essentially a non-functional proof of concept where two different types of visualizations are stitched together. A screenshot can be seen in Figure 10. The upper one is a view of the world map where each country is assigned a hue of color based on the percentage of queries that end up in a foreign country (a darker hue indicates a higher percentage), better known as a choropleth. When a country is selected, the bar chart shows the distribution per country. In this case 83.19% percent of the queries from Turkey (TR) would end up at an anycast site in Italy (IT).

One of the main points of critique was the use of a choropleth (also briefly touched upon in paragraph 3.4), since it tends to skew the user's attention to the countries with a larger surface area. Another side effect is that it becomes almost impossible to see the percentages of small countries such as Luxembourg.







### 5.2.2 Iteration 2

The second prototype has changed significantly in comparison to the previous iteration cycle. A screenshot can be seen in Figure 11. Amongst the main differences is the switch from a choropleth to a bubble chart for reasons that were found in the ideation phase and expanded upon in paragraph 3.4. It

was decided to assign each country a color using the RAG scale<sup>6</sup> (RAG standing for Red, Amber and Green), better known as traffic light colors, see Table 3. These colors are more intuitive than using a single color gradient used in the previous prototype.

Furthermore, numerous interaction techniques have been implemented. These include brushing and linking, zooming and details-on-demand. Brushing and linking allows you to select a country using the map. Once selected the bar chart then shows the data of that particular country. The world map supports zooming to get a closer look of an area of interest. Details-on-demand is implemented using a tooltip, which shows the actual percentage once you hover over a country using the mouse.

Finally, the two visualizations are now positioned next to each other instead of stacked vertically. Since the visualization is most likely to be viewed on a computer monitor, it makes sense to match the aspect ratio (16:9) such that the entire screen can be utilized.

	Color	Indicator	Description
R	Red	Problematic	Requires action
Α	Amber	lssues	Some aspects require attention, some good
G	Green	Good	All is good

#### Table 3 – Colors of the RAG scale

The overall feedback was positive. The added features were well received and the interaction possibilities seemed intuitive to use. Some features came to mind that may be useful to have, such as viewing the percentages of the queries that end up at an anycast site in a different continent. Whereas it previously only showed the percentage of the queries that end up in a foreign country. Another suggestion was the option to also show absolute values next to the relative values that were already shown.

<sup>&</sup>lt;sup>6</sup> The RAG scale is quite common in the food industry for product labeling, but is also used for performance monitoring



#### Figure 11 – Screenshot of prototype 2

#### 5.2.3 Iteration 3

The third prototype is a less drastic redesign in comparison to the second one, but still many changes have been made. A screenshot can be seen in Figure 12. In this version it is now possible to switch between the desired measure, either the percentage of the queries that go intercontinental or international. Furthermore, the tooltip has been redesigned to show the name of the country, both percentages for intercontinental and international and the total number of measurements that were taken in that country. The third feature that was added in this iteration cycles was the ability to filter on the continent of interest. Once a continent is selected, the map will automatically filter out the other continents and zoom in on the one that is selected. Finally, it was found that many countries did not show up in the visualization even though they were present in the dataset. This was due to a mistake made in the data preparation phase, which has now been resolved.

Several points of feedback were given. De Vries mentioned that some of the circles are not displayed in the center of each country, e.g. the one in Norway. This would be quite complicated to fix, as it would require manually modifying the latitude and longitude GPS coordinates for each country. Furthermore, a legend is missing for the map to explain what the different colors of the circles represent.



#### Figure 12 – Screenshot of prototype 3

#### 5.2.4 Iteration 4

The fourth iteration includes some aesthetic improvements and a few extra functionalities. A screenshot can be seen in Figure 13. Concerning the aesthetic changes, the measure selection and filters have been redesigned and relocated to the top and center of the screen. The titles of each element have been made easier to read and are now all in the same style. Moreover, a legend has been added and the bar chart now has titles on both axes. Furthermore, a somewhat smart filter has been implemented which, when enabled, will only show the countries that have an anycast site, yet a percentage of the queries from this country still go across the border. Next to this button to filter countries, two more buttons were added. The first is to reset all the selections and filters to the default state. The second button is to give the user additional information about the visualization in case this may be desired.

The visual overhaul and the new functionalities were well received, especially the filter to only show the countries with an anycast site was appreciated. The most important feedback given was that it is currently not possible to view the distribution per autonomous system, i.e. it's impossible to see from which Internet service provider the queries originate.





#### 5.2.5 Iteration 5

The fifth prototype includes a few extra, major features. Further changes to the prototypes are mostly aesthetically, although some small bugs have been squashed as well. A screenshot can be seen in Figure 14. In the feedback from the previous prototype an interest was expressed for the ability to see the distribution of queries per autonomous system. This has now been implemented. Where the previous versions had one bar chart, there are now two of them. The top one shows the distribution per continent, country, anycast site and Autonomous System Name, whereas the bottom one shows the distribution per Autonomous System Number. Furthermore, it's possible to search for specific countries and Autonomous Systems by name and to drill down in to the raw data. The titles are now dynamic instead of static, since they adapt to the filters and measures that have been selected to better inform the user about the data they are currently viewing. Finally, the visualization can now be accessed through a web page making it easier to share with others.

This is the version that was presented to SIDN to get more extensive feedback. The results of this feedback session and the process of a final iteration cycle can be found in the next chapter.





### 5.3 Conclusion

The goal of this phase was to implement the visualization in a structured manner using the results obtained in the previous phases. To prepare the data a combination of a Java program and an R script have been used. After the data was ready to be visualized multiple iteration cycles have been executed to implement the visualization. Each iteration improved upon the previous one and feedback was gathered from a stakeholder. After prioritizing the received feedback it was used to build the next prototype. In total 5 iteration cycles were completed where the fifth prototype will be used to evaluate with a larger group of stakeholders, which will be discussed in the next chapter.

# 6 Evaluation

After the fifth iteration of prototyping, it is time to obtain feedback from various stakeholders (other than de Vries). This was done by pitching and demonstrating the prototype to the people that were present during a meeting of the SAND project<sup>7</sup>, which is further explained in paragraph 6.1. Subsequently, the received feedback will be discussed after which the final iteration cycle will be presented.

# 6.1 SAND project

SAND stands for Self-managing Anycast Networks for DNS and is a joint project from the University of Twente, SIDN and NLnet Labs. Quoting from their website:

"The SAND team performs applied research in Anycast services aiming to provide tools and recommendations for DNS operators. The goal is to develop, prototype, and evaluate an intelligent and measurement-based recommendation tool for operators of anycast services..."

The SAND project team mainly consists out of researchers and network operators employed at the organizations that were mentioned above.

# 6.2 Results

At the meeting the prototype was pitched and demonstrated to the people present. This already resulted in an open discussion where various remarks were made and things were discussed. Afterwards they also have been asked to think of possible improvements with regard to the functionality (e.g. the usefulness, interaction support and data granularity) and design (e.g. attractiveness and chart types). From this feedback session quite some useful comments were received:

- The first issue was that it is not possible to see to which continent a country belongs. For the well-known countries this is not such a problem, e.g. countries within Europe. However, for the lesser known countries, where it is not as obvious to which continent it belongs such as countries in the middle of the Atlantic Ocean, it can be useful to be able to see its corresponding continent.
- Secondly, it was suggested to make use of the Round Trip Time (RTT) between the users and the anycast sites. In the ideal situation this metric would have been used in the first place, because

<sup>&</sup>lt;sup>7</sup> http://www.sand-project.nl/about/

it arguably gives the best indication of how well of a choice was made by the BGP to select a particular anycast site for a given user. However, this kind of data is not available. Currently there is being worked on to extend the Verfploeter tool to also collect RTT's, but the results so far have not been reliable and accurate.

- Furthermore, an idea was put forward to visualize the differences between two datasets. Currently the visualization makes use of a single dataset obtained through a measurement of the Verfploeter tool. Doing two measurements with some time in between would allow you to see the differences over time. When the differences between the datasets are calculated on beforehand in the R script, the same or a very similar design and lay-out of the visualization could be used to visualize this data.
- Some concerns were expressed about using Microsoft Power BI as a platform. He thought it
  would not fit in to their workflow, because it is a closed source application and thus uses
  proprietary software. Using low-level, open source software to build the visualization would
  have had the preference for the final application. This would however require much more time
  for each iteration cycle and would take even longer before a useable prototype can be
  presented, as was discussed in paragraph 2.3.
- Finally, someone proposed to also use the amount of measurements that are taken in each country as a measure that can be selected and thus to be represented in the world map. Right now it is only possible to select the percentage of queries that go either intercontinental or international as a measure.

# 6.3 Final iteration

In the final iteration cycle of prototyping several important changes have been made based on the feedback that was received at the SAND project meeting. A screenshot can be seen in Figure 15. First of all, when hovering over a country the tooltip now shows to which continent this country belongs. Once the country has been selected, its corresponding continent can now also be seen in the titles of the two bar charts. Furthermore, a new measure has been added to the two that were already present. When activated the world map will visualize the amount of measurements that have been taken for each country. Finally, the code has been cleaned up and some small design tweaks have been made.





### 6.4 Conclusion

The goal of this phase was to obtain as much constructive feedback as possible in order to do a final iteration of the prototype. The overall feedback received at the SAND meeting was quite positive. People liked the design and were fairly enthusiastic about the visualization. Some small issues came to light, such as not being able to see the continent to which a country belongs or not being able to select the number of measurements as a measure. These were overcome in the final iteration. The main concern was about the visualization being built using Microsoft Power BI as it is not an open source framework, which was more or less expected and discussed in paragraph 2.3.

# 7 Conclusion

A high-fidelity prototype of an interactive data visualization has been realized that can visualize how an anycast network is functioning. Numerous interaction techniques have been utilized, which all have been discussed in the literature review in paragraph 2.2, to allow the user to view, explore and analyze the data. Amongst others, the support to drill down in to the data to see the distribution per continent, country, anycast site and autonomous system contributes to this.

The main research question was how to implement an interactive data visualization that can visualize the functioning of an anycast network. Crucial to answering this was specifying the system requirements. This was an ongoing process throughout the entire project and already started during the background research where amongst other things related visualizations were inspected. With these in mind different designs were made that have been evaluated with de Vries to obtain feedback. Subsequently, the human-centered design process played an important role, which involved doing multiple iteration cycles. The feedback received after each evaluation session was prioritized accordingly and was taken in to account for the next iteration cycle. The second last version was used for a more extensive feedback session at a SAND project meeting after which some last improvements have been made and a final prototype was presented.

Even though some concerns were expressed about the final prototype being built in Microsoft Power BI, the prototype can serve as a framework and starting point for future development. A foundation has been set on which can be built upon as will be further discussed in chapter 9.

# 8 Discussion

First of all, the data preparation phase took more time than expected where only figuring out the right data structure was already quite time consuming. The actual structuring of the data using R went without too many complications, although one mistake was made: A large percentage of the countries were accidentally filtered out of the dataset, but this was quickly found out and resolved in the next iteration.

A point of discussion could be the accuracy of the GeoIP2 databases that were used to retrieve the country of origin and Autonomous System for each IP address. According to their website<sup>8</sup>, the accuracy on a country level is 99.8%, which is excellent. For the AS this lies around the 80%, which is still adequate, but not too convincing and should therefore be used with care. Even worse is the accuracy on a city level, which is only 38% in the Netherlands in the free version of the database. Even though the commercial version has a slightly better accuracy of 49% in the Netherlands, this is still not acceptable. For this reason it was chosen not to make use of this database.

Furthermore, no user tests were executed for this project. This is because it's rather troublesome to find representative users that have the time and are willing to cooperate in a user test. An alternative would have been to run the tests on students. Although, the results would presumably be unreliable as a student doesn't have the same knowledge a network operator or a researcher in this area of expertise has.

Finally, most of the things only work on Windows, i.e. not on Linux or macOS. So, if changes need to be made to the visualization Microsoft Power BI must be used. Also the batch file that automates the data preparation only works on Windows. Although, this could easily be translated to a shell script, which does run on Linux or macOS.

Some of the points mentioned above will be continued upon in the next section as it discusses some suggestions for future work.

<sup>&</sup>lt;sup>8</sup> Accuracy of the GeoIP2 databases – https://www.maxmind.com/en/geoip2-country-database

# 9 Future work

A high-fidelity prototype of an interactive data visualization has been realized. Throughout the process leading to the end result multiple functionalities came to mind that have not been implemented. Reason being that adding the functionality was either not feasible due to time constraints or it was simply not possible at that moment. Now follows an enumeration of ideas or suggestions for future work:

- This prototype makes use of the free version of Maxmind's GeoIP2 service (to retrieve locational data for all IP addresses). They also provide a commercial version, which can be purchased on their website<sup>9</sup>. The latter one is updated more often and has a better accuracy than the free version. Although, the accuracy on a city level is not yet adequate at the time of writing, as discussed in chapter 8, it may improve over time and could thus be a decent improvement to the visualization at some point.
- In this version three different measures have been implemented, but there are more that can be added whenever this data becomes available. Measures you could think of are e.g. the RTT, Time To Live (TTL) or the hop count between the user and server.
- Doing multiple measurements of the same anycast network would make it possible to visualize how it develops over time. This would allow you to detect certain events, such as the reconfiguration of a routing policy. The visualization could be used to find out whether this had a positive or negative effect on the network's performance.
- Switching to another dataset to be used in the visualization is currently done manually on a local computer on which the dataset is located. It would be more practical to store the dataset on a server to which the visualization is connected. This way it could refresh automatically as soon as the new dataset is uploaded to the server.
- The Java program can only handle datasets that have the same structure otherwise it will fail to retrieve the additional data accompanying with each IP address. To prevent this a function could be added that can detect the structure of the dataset to ensure the data retrieval succeeds.

<sup>&</sup>lt;sup>9</sup> Maxmind commercial databases – https://www.maxmind.com/en/geoip2-databases

# Appendix

# A1 – Pen-and-paper prototypes





### A2 – Java program

//author: Cas Wiefferink
//retrieves country, ASN and ASN-name of an IP Address
//using GeoIP2 (created by MaxMind, available from www.maxmind.com) and OpenCSV
//input IP addresses and results are respectively read from and written to a csv file

```
import com.maxmind.geoip2.DatabaseReader;
import com.maxmind.geoip2.model.AsnResponse;
import com.maxmind.geoip2.model.CountryResponse;
import com.opencsv.CSVReader;
import com.opencsv.CSVWriter;
```

```
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.net.InetAddress;
import java.util.Arrays;
```

```
public class Main {
```

```
public static void main(String[] args) {
```

```
System.out.println(calcDistance(32.9697, -96.80322, 29.46786, -98.53506));
        try {
            getCountryAndASN();
        } catch (IOException e) {
           e.printStackTrace();
        }
    }
   public static void getCountryAndASN() throws IOException {
        CSVReader reader = new CSVReader(new FileReader("Datasets\\dataset.csv"), ',',
'"', 1);
        String csv = "Datasets\\output.csv";
        CSVWriter writer = new CSVWriter(new FileWriter(csv));
        // open country database
        String dbCountry = "Datasets\\GeoLite2-Country.mmdb";
        File countryDatabase = new File(dbCountry);
        DatabaseReader countryReader = new
DatabaseReader.Builder(countryDatabase).build();
        // open ASN database
        String dbASN = "Datasets\\GeoLite2-ASN.mmdb";
        File asnDatabase = new File(dbASN);
        DatabaseReader asnReader = new DatabaseReader.Builder(asnDatabase).build();
        //write headers to csv file
        String[] header =
("IP Address;Country;ASN Number;ASN Name;Anycast site").split(";");
        writer.writeNext(header);
        int count = 0;
        int count2 = 0;
        //Read CSV file line by line
        String[] nextLine;
        while ((nextLine = reader.readNext()) != null) {
            if (nextLine != null) {
                //parse to string
                String data = Arrays.toString(nextLine);
```

```
String[] row = data.split(",");
                //remove space before each entry
                row[1] = row[1].substring(1);
                row[2] = row[2].substring(1);
                //only use entry when the 2 IP addresses (destination and source)
match
                if (row[1].equals(row[2]) == false) {
                    count++;
                } else {
                    row[6] = row[6].substring(1);
                    try {
                        //translate to an IP Address
                        InetAddress ipAddress = InetAddress.getByName(row[1]);
                        //create response objects for country and asn
                        CountryResponse countryResponse =
countryReader.country(ipAddress);
                        AsnResponse asnResponse = asnReader.asn(ipAddress);
                        //retrieve country and asn using response objects
                        String countryName = countryResponse.getCountry().getName();
                        String asnName =
asnResponse.getAutonomousSystemOrganization();
                        int asnNumber = asnResponse.getAutonomousSystemNumber();
                        //format and write to csv file
                        String[] entries = (row[1] + ";" + countryName + ";" +
asnNumber + ";" + asnName + ";" + row[6]).split(";");
                        writer.writeNext(entries);
                        //no information about IP address was found--> skip and
continue
                    } catch (Exception exception) {
                        count2++;
                        continue;
                    }
                }
            }
        }
        //print the amount of exceptional cases
        System.out.println(count + " cases where destination- and source IP
differed");
        System.out.println(count2 + " cases where no information about the IP was
found");
        writer.close();
    }
}
//author: Cas Wiefferink
//calculates the distance between two points using their latitude and longitude
package com.company;
        import java.lang.*;
public class DistanceCalculator {
    //requires latitudes and longitudes of the two locations
   public static double calcDistance(double lat1, double lon1, double lat2, double
```

lon2) {

```
//if locations are equal, return 0
        if ((lat1 == lat2) && (lon1 == lon2)) {
           return 0;
        } else {
            //some math
            double theta = lon1 - lon2;
            double dist = Math.sin(Math.toRadians(lat1)) *
Math.sin(Math.toRadians(lat2)) + Math.cos(Math.toRadians(lat1)) *
Math.cos(Math.toRadians(lat2)) * Math.cos(Math.toRadians(theta));
            dist = Math.acos(dist);
            dist = Math.toDegrees(dist);
            dist = dist * 60 * 1.1515 * 1.609344;
            return (dist);
        }
   }
}
```

### A3 – R Script

#set working directory setwd("E:/Downloads/icmp measurement 20170201/Version 1.6") *#using dplyr and magrittr* install.packages("dplyr") library("dplyr") install.packages("magrittr") library(magrittr) #import datasets from folder called "Datasets" in working directory codesIATA <- read.csv("Datasets/codesIATA.csv")</pre> countries <- read.csv("Datasets/country\_ISO.csv")</pre> output <- read.csv("Datasets/output.csv")</pre> continents <- read.csv("Datasets/continents.csv", stringsAsFactors = FALSE)</pre> continents2 <- read.csv("Datasets/continents Power BI.csv", stringsAsFactors = FALSE) #remove rows of IATA dataset where municipality is NA or iata code is 0, remove latitude and longitude columns codesIATA <- codesIATA %>% filter(municipality != "", iata code != 0) %>% select(ntiso country, municipality, iata code) #remove latitude and longitude columns countries <- countries %>% select(country, name) #remove entries where country was not found and remove duplicates output <- filter(output, Country != "null")</pre> output <- output %>% distinct(IP Address, .keep all = TRUE) #merge datasets to retrieve the country for each Anycast site output2 <- merge(x=output, y=codesIATA, by.x = "Anycast site", by.y = "iata code", all.x = TRUE) *#remove unwanted columns* output2\$municipality <- NULL #rename columns to make them easier to understand output2 <- rename(output2, Country\_IP=Country)</pre> output2 <- rename(output2, Country\_ISO\_Anycast\_Site=ntiso\_country)</pre> output2 <- rename(output2, Anycast Site=Anycast site)</pre> #merge datasets to retrieve the full name of the anycast site country output3 <- merge(x=output2, y=countries, by.x = "Country ISO Anycast Site", by.y = "country", all.x = TRUE) #rename columns to make them easier to understand output3 <- rename(output3, Country Anycast Site=name)</pre> output3\$Country ISO Anycast Site <- NULL #calculate occurrences for each combination of IP Country, Anycast site and ASN anycast\_per\_country <- output3</pre> anycast\_per\_country <- anycast\_per\_country %>% group\_by\_all() %>% count

#calculate percentage for each combinaton of Anycast Country, IP Country and ASN anycast per country <- group by (anycast per country, Country IP) %>% mutate (percent = n/sum(n)#remove unwanted column and rename column to make it easier to understand anycast per country\$ASN Name <- NULL anycast per country <- rename (anycast per country, Country Anycast=Country Anycast Site) #reorder columns anycast per country <- anycast per country[c(1,2,5,6,4,3)] #calculate occurrences AGAIN, now for each combinaton of Anycast Country and TP Country combination Count <- count(output3, Country Anycast Site, Country IP) #check whether sum of all the frequencies equeals the amount of observations sum(combination Count\$n) *#calculate percentage for each combinaton of Anycast Country and IP Country* combination Percentage <- group by (combination Count, Country IP) %>% mutate (percent = n/sum(n)#new table for the percentage that ends up in a foreign country #when there are also queries that do NOT leave the country (-> thus knowing the country has a anycast site for itself) percentage Foreign <- subset(combination Percentage,</pre> as.character(Country\_Anycast\_Site) == as.character(Country\_IP)) percentage Foreign\$Country Anycast Site <- NULL #new table for the percentage that ends up in a foreign country, for ALL the countries percentage Foreign2 <- subset(combination Percentage,</pre> as.character(Country Anycast Site) != as.character(Country IP)) percentage Foreign2 <- percentage Foreign2 %>% group\_by(Country IP) %>% summarise(n = sum(n), percent = 1-sum(percent)) percentage\_Foreign2 <- union(percentage\_Foreign2, percentage\_Foreign)</pre> percentage\_Foreign2 <- percentage\_Foreign2 %>% group by (Country IP) %>% summarise(n = sum(n), percent = 1-mean(percent)) *#write to .csv files* #write.csv(combination Percentage, file = "Output/country country perc.csv") write.csv(anycast per country, file = "Output/country anycast perc2.csv") #write.csv(percentage Foreign2, file = "Output/country foreign perc.csv") ###################calculate percentage intercontinental########################## continents <- continents %>% select(country, continent) *#union two continent datasets* continents <- union(continents, continents2)</pre> continents <- continents %>% distinct(country, .keep all = TRUE) #retrieve continent for both country\_IP and country\_anycast combination Percentage <- merge(x=combination Percentage, y=continents, by.x = "Country IP", by.y = "country", all.x = TRUE) combination Percentage <- rename(combination Percentage, Continent IP=continent) combination Percentage <- merge(x=combination Percentage, y=continents, by.x = "Country Anycast Site", by.y = "country", all.x = TRUE) combination Percentage <- rename(combination Percentage,</pre> Continent Anycast Site=continent)

#filter entries where Country or Continent was not found combination Percentage <- filter(combination Percentage, Country Anycast Site != "null") combination Percentage <- filter(combination\_Percentage, Continent\_IP != "null")</pre> #new table for the percentage that ends up in a foreign country #when there are also queries that do NOT leave the country (-> thus knowing the country has a anycast site for itself) percentage\_Intercontinental <- subset(combination\_Percentage,</pre> as.character(Continent Anycast Site) == as.character(Continent IP)) # percentage Foreign\$Country Anycast Site <- NULL</pre> #new table for the percentage that ends up intercontinental, for ALL the countries percentage Intercontinental2 <- subset (combination Percentage, as.character(Continent Anycast Site) != as.character(Continent IP)) percentage Intercontinental <- percentage Intercontinental %>% group by (Country IP) %>% summarise(n = sum(n), percent = 1-sum(percent)) percentage Intercontinental2 <- percentage Intercontinental2 %>% group\_by(Country\_IP) %>% summarise(n = sum(n), percent = sum(percent)) percentage Intercontinental2 <- union (percentage Intercontinental2, percentage Intercontinental) percentage Intercontinental2 <- percentage Intercontinental2 %>% group\_by(Country\_IP) %>% summarise(n = sum(n), percent = mean(percent)) #combining international and intercontinental in to one dataset and filter missing entries percentage Foreign2 <- rename(percentage Foreign2, international=percent)</pre> percentage Intercontinental2 <- rename (percentage Intercontinental2, intercontinental=percent) inter\_National\_Continental <- merge(x=percentage\_Foreign2,</pre> y=percentage Intercontinental2, by.x = "Country IP", by.y = "Country IP", all.x = TRUE) inter National Continental <- filter(inter National Continental, intercontinental != "null") #renaming a column and removing one duplicate column inter National Continental <- rename(inter National Continental, n=n.x) inter National Continental <- rename(inter National Continental,</pre> percent=international) inter National Continental <- inter National Continental %>% select(Country\_IP, n, percent, intercontinental) #write to .csv file write.csv(inter National Continental, file = "Output/inter national continental.csv") *#contintents lookup table* continents <- read.csv("Datasets/airport-codes\_csv.csv", stringsAsFactors = FALSE)</pre> continents <- continents %>% select(continent, iso country) %>% distinct(iso\_country, .keep\_all = TRUE) continents <- merge(x=continents, y=countries, by.x = "iso country", by.y = "country", all.x = TRUE) continents\$iso country <- NULL

```
#replace continent abbreviations with full name
continents <- continents %>%
  mutate(continent = replace(continent, continent == "EU", "Europe")) %>%
  mutate(continent = replace(continent, continent == "SA", "South America")) %>%
  mutate(continent = replace(continent, which(is.na(continent)), "North America")) %>%
  mutate(continent = replace(continent, continent == "AF", "Africa")) %>%
  mutate(continent = replace(continent, continent == "AS", "Asia")) %>%
mutate(continent = replace(continent, continent == "OC", "Oceania")) %>%
mutate(continent = replace(continent, continent == "AN", "Antarctica"))
#remove entries with missing country and rename country column
continents <- rename(continents, country=name)</pre>
continents <- filter(continents, country != "null")</pre>
write.csv(continents, file = "Output/continents.csv")
write.csv(continents, file = "Output/continents - Copy.csv")
#codesIATA lookup table
codesIATA$ntiso_country <- NULL</pre>
codesIATA <- rename(codesIATA, anycast_site=iata_code)</pre>
codesIATA <- distinct(codesIATA, anycast_site, .keep_all = TRUE)</pre>
write.csv(codesIATA, file = "Output/codesIATA.csv")
#AS lookup table
ASN <- distinct (output3, ASN Number, .keep all = TRUE)
ASN <- ASN %>%
  select(ASN_Number, ASN_Name)
write.csv(ASN, file = "Output/ASN.csv")
```

# References

Aben, E., Buddhdev, A., & Petrie, C. (2017). DNS Root Server Transparency: K-Root, Anycast and More — RIPE Labs. Retrieved October 15, 2018, from https://labs.ripe.net/Members/emileaben/dns-root-server-transparency

Abley, J., & Lindqvist, K. (2006). Operation of Anycast Services. https://doi.org/10.17487/rfc4786

Clegg, D., & Barker, R. (1994). *Fast-track : a RAD approach*. Addison-Wesley Pub. Co. Retrieved from https://dl.acm.org/citation.cfm?id=561543

Cloudflare. (n.d.). What is BGP Hijacking? Retrieved April 11, 2019, from https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/

Craft, B., & Cairns, P. (2005). Beyond guidelines: What can we learn from the visual information seeking mantra? *Proceedings of the International Conference on Information Visualisation*, 2005, 110–118. https://doi.org/10.1109/IV.2005.28

de Vries, W. B., de O. Schmidt, R., Hardaker, W., Heidemann, J., de Boer, P.-T., & Pras, A. (2017). Broad and load-aware anycast mapping with verfploeter. *Proceedings of the 2017 Internet Measurement Conference on - IMC '17*, *17*, 477–488. https://doi.org/10.1145/3131365.3131371

Demchak, C., & Shavitt, Y. (2018). China's Maxim – Leave No Access Point Unexploited: The Hidden Story of China Telecom's BGP Hijacking. *Military Cyber Affairs*, *3*(1). https://doi.org/10.5038/2378-0789.3.1.1050

Dix, A., & Ellis, G. (1998). Starting Simple - adding value to static visualisation through simple interaction. *Advanced Visual Interfaces AVI98*. (pp. 124-134). ACM. https://doi.org/10.1145/948496.948514

Doleisch, H. (2004). *Visual Analysis of Complex Simulation Data using Multiple Heterogeneous Views*. (PhD thesis) Retrieved from https://www.cg.tuwien.ac.at/research/publications/2004/doleisch-thesis/

Figueiras, A. (2015). Towards the understanding of interaction in information visualization. *Information Visualisation (iV), 2015 19th International Conference* (pp. 140-147). https://doi.org/10.1109/iV.2015.34

International Organization for Standardization. (2010). Ergonomics of human-system interaction: Human-centred design for interactive systems : ISO 9241-210. *International Organization for Standardization*, 2010, 32. Retrieved from https://www.iso.org/standard/52075.html

Jensen, C. S., Pedersen, T. B., & Thomsen, C. (2010). *Multidimensional Databases and Data Warehousing*. *Synthesis Lectures on Data Management* (Vol. 2). Morgan & Claypool Publishers. Available from https://doi.org/10.2200/S00299ED1V01Y201009DTM009

Keim, D. A. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics* (Vol. 8). https://doi.org/10.1109/2945.981847

Keim, D. A., Mansmann, F., Schneidewind, J., & Ziegler, H. (2006). Challenges in Visual Data Analysis. *Tenth International Conference on Information Visualisation (pp. 9-16)* https://doi.org/10.1109/IV.2006.31

Khan, M., & Khan, S. (2011). Data and information visualization methods, and interactive mechanisms: A survey. *International Journal of Computer Applications 34*(1), 1-14 Retrieved from https://www.researchgate.net/publication/264623537\_Data\_and\_Information\_Visualization\_Methods\_and\_Interactive\_Mechanisms\_A\_Survey

Kohlhammer, J., Keim, D., Pohl, M., Santucci, G., & Andrienko, G. (2011). Solving Problems with Visual Analytics. *Procedia Computer Science*, *7*, 117–120. https://doi.org/10.1016/J.PROCS.2011.12.035

Kosara, R., Hauser, H., & Gresh, D. L. (2003). An Interaction View on Information Visualization. *Eurographics*, 123–137. Retrieved from https://cdn.mprog.nl/dataviz/excerpts/w5/Kosara\_Interaction\_View.pdf

Pillai, S. (2014). What is Anycast and How it works. Retrieved October 13, 2018, from https://www.slashroot.in/what-anycast-and-how-it-works

RIPE NCC Staff. (2015). RIPE Atlas: A Global Internet Measurement Network. *The Internet Protocol Journal*, *18*(3), 2–26. https://doi.org/10.1007/s10393-014-0961-8

Rowley, J. (2007). The wisdom hierarchy: Representations of the DIKW hierarchy. *Journal of Information Science*, *33*(2), 163–180. https://doi.org/10.1177/0165551506070706

Sallam, R. L., Howson, C., Idoine, C. J., Oestreich, T. W., Laurence Richardson, J., & Tapadinhas, J. (2017). *Magic Quadrant for Business Intelligence and Analytics Platforms*. Retrieved from https://cdn2.hubspot.net/hubfs/2172371/Q1 2017 Gartner.pdf?t=14962606

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Visual Languages, 1996. Proceedings., IEEE Symposium on (pp. 336-343)*. https://doi.org/10.1016/B978-155860915-0/50046-9

van Bruxvoort, X. (2018). *The impact of #MeToo; a data visualization*. Retrieved from https://essay.utwente.nl/75556/1/van\_Bruxvoort\_Creative\_Technology\_EEMCS.pdf

von Huhn, R. (1927). Further Studies in the Graphic Use of Circles and Bars. Journal of the American Statistical Association (Vol. 22). https://doi.org/10.1080/01621459.1927.10502938

Wilkinson, L. (2006). *The Grammar of Graphics* (Second Edi). Springer Science & Business Media. https://doi.org/10.1007/0-387-28695-0

Yi, J. S., Kang, Y. A., Stasko, J., & Jacko, J. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization & Computer Graphics, (6).* https://doi.org/10.1109/TVCG.2007.70515