DOI: 10.1111/cgf.14287
Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

ProBGP: Progressive Visual Analytics of Live BGP Updates

A. Ulmer[1], D. Sessler[1] and J. Kohlhammer[1,2]

[1]Fraunhofer IGD, Germany
[2]Technische Universität Darmstadt, Germany

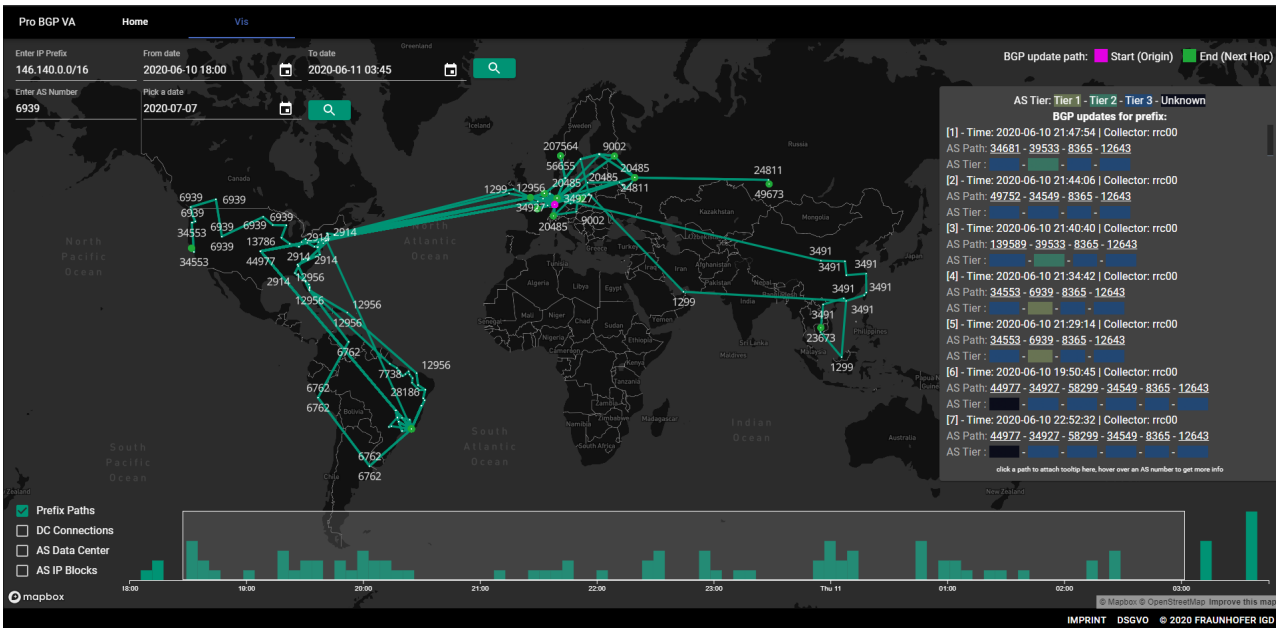

Figure 1: *ProBGP visual interface for BGP update queries and geographic visualization: The main visualization is a map that is overlayed with query input fields in the top left. There the user can insert an AS number and a date to get the geographic approximation of the AS data centers. This figure shows routes for 146.140.0.0/16 for about 10 hours on the 10th and 11th of June 2020. The pink dot shows the origin AS and green dots the next hop after a collector recorded a BGP update. The table on the right is displayed as the user hovers or selects update paths. More details about an AS can be requested by hovering an ASN in the table. The coloring shows the AS tier.*

Abstract

The global routing network is the backbone of the Internet. However, it is quite vulnerable to attacks that cause major disruptions or routing manipulations. Prior related works have visualized routing path changes with node link diagrams, but it requires strong domain expertise to understand if a routing change between autonomous systems is suspicious. Geographic visualization has an advantage over conventional node-link diagrams by helping uncover such suspicious routes as the user can immediately see if a path is the shortest path to the target or an unreasonable detour. In this paper, we present ProBGP, a web-based progressive approach to visually analyze BGP update routes. We created a novel progressive data processing algorithm for the geographic approximation of autonomous systems and combined it with a progressively updating visualization. While the newest log data is continuously loaded, our approach also allows querying the entire log recordings since 1999. We present the usefulness of our approach with a real use case of a major route leak from June 2019. We report on multiple interviews with domain experts throughout the development. Finally, we evaluated our algorithm quantitatively against a public peering database and qualitatively against AS network maps.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;*** • ***Networks*** → *Data center networks;* • ***Theory of computation*** → *Routing and network design problems;*

1. Introduction

Interdomain routing is the core of the Internet. Autonomous systems (ASes) manage the routing of packets from your device to the service you are using by exchanging routing information via the Border Gateway Protocol (BGP). They exchange, to which block of IP addresses (also called IP prefix) they can provide a route. The analysis of the routing network has a long history in research and is an important domain for Internet Service Providers (ISPs). Every year routing misconfigurations or attacks on the insecure BGP result in intercontinental outages that afflict major services like Google or WhatsApp [Tho20a, Tho20b]. However, there are many more small anomalies, which are not detected or made public, because there is no log data for the complete routing network. There are only three projects, which installed 59 collectors at ASes around the world to gather log data. Although this is not covering the complete routing network, the log data generated by these collectors is already too large to analyze it entirely. In the past, multiple applications were proposed to approach this problem (see Section 2.1) but none of them focused on a high accuracy geographic visualization of routing paths. If a path between ASes does not take the shortest path geographically it may be an indicator for anomalous behavior. ASes are divided into three tiers, which reflect their global connectivity to other ASes, where Tier 1 ASes are global networks providing a connection between smaller Tier 2 networks which do the same for Tier 3 networks. The problem is that ASes are not present in only one location but they are distributed systems with multiple data centers. There is almost no data about where these data centers are and how they are connected internally because most of the ISPs do not want to share this business information. However, there are GeoIP databases that offer approximate locations for IP addresses and information about which IP addresses belong to each AS. Based on this data, we developed a novel approximation algorithm for AS data centers and their internal connections. With that, we are able to visualize the data centers on a world map and compute the geographic paths of BGP updates. We created a progressive visual analytics application to analyze live BGP update logs. We evaluated our prototype with the main user groups which play a role in this domain. Our prototype is available at https://probgp.igd.fraunhofer.de and works best with Google Chrome. Our main contributions are:

1. Geospatial approximation of internal AS networks based on geographic distance and managed IP addresses
2. Always up-to-date, progressive visual analytics application for BGP update path analysis
3. Insights of how visualization can contribute to the BGP domain, based on result validations and evaluations with domain experts

The paper is structured as follows: In Section 2, we discuss the related work on network visualization and progressive visual analytics. Section 3 explains the different user groups and their tasks as well as the data types that we use. We explain the approximation algorithm and our progressive visual analytics approach in Sections 4 and 5. In Section 6 we show how our prototype can solve the previously defined tasks with a real-world use case. We present the results of our evaluation in Section 7, before concluding with a discussion of challenges, limitations, and future work.

2. Related Work

Our visual analytics approach (as defined by Keim et al. [KKEM10]) combines data management and automated analysis with visualization. It is further inspired by previous work in network visualization and progressive analytics. Of particular interest for this paper are geospatial network visualization, BGP update visualizations, and guidelines for progressive approaches that allow online access with responsive user interaction.

2.1. Network Visualization

Network and graph visualizations [BEW95] are certainly among the most prominent and widely used information visualizations with a wide variety of applicable data sources and application domains ranging from social networks [HB05], over traffic analysis [LAA*20] to cybersecurity [Cam21]. This paper is especially interested in geospatial network visualization with still a wealth of related work, techniques, and systems. A recent survey by Schöettler et al. analyzes 95 such visualizations and provides a classification scheme [SYPB21]. Within cybersecurity as an application domain, network visualization and in particular routing visualization has a long history because of the interesting problem to find anomalies in the very large amounts of raw data. Many early approaches for BGP routing (e.g. [DBMPP03]) visualized the AS network in a node-link diagram where the nodes are the ASes and the links represent the BGP update paths. But without domain expert knowledge it is difficult to understand if a routing change is suspicious.

Several approaches have argued that it is easier to detect an anomalous change if the BGP update path is visualized geographically. Following the above classification [SYPB21], we are particularly looking at interactive, super-imposed geospatial network visualizations with a geographically mapped, explicit representation of nodes and links. Syamkumar et al. [SDB16] used their system Bigfoot to preprocess BGP update logs for one year and found 139 candidate events which they visualize. They render a 2D polygon on the world map for each IP prefix that belongs to an AS, which results in many polygons for the coverage area of the AS. Although this approach provides a fast indication of anomalous behavior it may suffer from overplotting when analyzing an AS with global coverage. Another related work is BGPViewer by Papadopoulus et al. [PMT13] which, similar to our approach, visualizes an AS graph on the world map. They visualize the ASes on the country-level and highlight the gained and lost IP ownership over a specified time frame, which makes irregular events like route leaks visible. The visual analytics tool VisTracer [FFV*12] analyzes traceroutes to find routing anomalies. Their focus is on the detection of anomalies but they also provide a rough geographical visualization on the country-level. All three related works have a low geographic accuracy or do not account for the factor that there are global ASes that cannot be assigned to a specific country. With our approach, we increase the geographic accuracy to the city-level, which makes it possible to find unusual routing paths even in smaller regions.

2.2. Progressive Visual Analytics

Progressive visual analytics is a research field that gained increasing attention over the past years. Due to ever-larger datasets, al-

gorithms take more time to compute final results. Thus, visual updates with intermediate results are necessary to keep the user involved in the analysis process. Earlier contributions to this field called this method incremental visualization [FPD*12] or progressive refinement [RS09] and already showed how the processing of large datasets can be handled. The term progressive visual analytics was established in the community by multiple publications [SPG14, FP16]. In a recent survey Angelini et al. [ASSS18] review all previous visual analytics contributions with progressive features and categorize them. Angelini et al. define two ways of progressiveness:

- *data chunking* - the partial results show increasingly more data over time.
- *process chunking* - the partial results show all the data at all time but the quality of approximation increases.

For data chunking the data source has to be sampled or sorted in a way that either the most important results are shown early or a given order in the data is preserved. An example for sampled data loading is the work of Badam et al. [BEF17] where they load large Twitter datasets and sample the dataset to enable a fast and progressive visualization. Then the data is processed iteratively and the results are presented to the user progressively after each iteration. For process chunking the whole data is processed, but the algorithm is able to produce intermediate results that can be visualized while the algorithm is progressively improving the accuracy of the result. A good example for this is a classifier, which is labeling the full dataset while it is progressively trained with more data, thus increasing the accuracy of the classification over time [PLvdM*16]. However, there are still many open research questions in this field, which were summarized by Fekete et al. [FFNS19]. We explain in detail how we implemented progressively in Section 5.1 which is highly dependent on the data and tasks defined in the next section.

3. Characterization of Data, Users and Tasks

In this section we specify the user groups and their tasks when working with BGP update data. First we describe all data sources in detail, then we introduce the main user groups interested in the analysis of this data and what their tasks are. Based on that, we define requirements for our prototype.

3.1. Data

We make use of five different data sources. We use GeoIP and AS-IP data to retrieve locations for IP blocks and the ASes that own these IP blocks. We use ASRank and Routing Information Service (RIS) data to enrich the information about ASes. Finally, we use BGP updates, which is by far the largest part of the data.

GeoIP and AS-IP Data

GeoIP data is an approximation of latitude and longitude coordinates for certain IP blocks. Multiple organizations gather this data and provide it for free with lower accuracy or as a paid service with higher accuracy. Following the related work on accuracy of GeoIP databases [KVR17, GSH*17], we chose the commercial GeoIP2 City database by Maxmind. We also chose Maxmind because they provide an AS-IP database, which includes the ownership of IP prefixes by AS number (ASN). With that we can compute all locations of IP blocks for a given ASN. But the IP blocks are not the same in both databases. For example, one large IP block owned by one AS might consist of multiple smaller IP blocks with different locations. Since we need to merge the two databases we use the merging algorithm developed by Ulmer et al. [USSK18]. During the analysis of the GeoIP data, we found that Maxmind has backup locations in many countries if they are not certain about the location of an IP block. They place these backup locations in water bodies (lakes, rivers) in the middle of a country or a larger city. This prevents confusion with other more accurate locations, but created problems for our algorithm. Currently we can only provide GeoIP data from June 2020 and will add new data as it is available. Requests to our BGP path visualization for time frames where GeoIP data is missing use the closest available data.

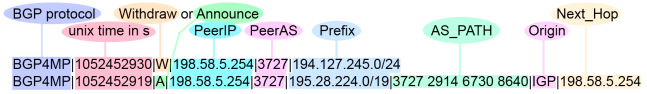

Figure 2: *Border Gateway Protocol Update Log Example*

BGP Updates

BGP updates are announcements or withdrawals of IP prefixes owned by an AS in the format shown in Figure 2. With those messages an AS communicates that all traffic for a certain IP address range should be routed to their data centers. Neighboring ASes get this message and update their routing table accordingly, add their AS number to the AS path and pass the message on to their neighboring ASes. Thus, each AS in the world learns where the next hop for a packet with a certain IP address is. Two scientific projects (Routeviews and Isolario) and the European network coordination centre (RIPE NCC) distributed 59 collectors around the world to gather data from the currently approximately 97k ASes. The data gathered at these collectors is dumped every five minutes for RIPE and Isolario collectors and every 15 minutes for Routeviews collectors. Updates have an average frequency of 240k with peaks up to 7 million in only 15 minutes. The collectors of RIPE produce on average 616 MB of data per collector each day in highly compressed archives (∼90%). For example just the collector rrc00 by RIPE gathered 44 GB compressed log data in June 2020, which is approximately 270 GB of uncompressed text data. From this data we mainly analyze the *as_path* but we also use the *time*, *announce/withdraw*, *prefix* and *next_hop* fields in our data processing.

AS Rank and RIS Data

ASes all have an unique number, their ASN. ASes are commonly divided into three tier levels, depending on their status as customer or provider of traffic. However, this is more of a well-accepted agreement than a binding definition. Tier 1 ASes are global networks which can reach all other networks without paying for the transit. Tier 3 ASes are small local networks which pay for transit and peering to reach other networks. And Tier 2 ASes are in the middle with a mixture of paid and free transit/peering. Luckie et al. [LHD*13] analyzed the relationships between the ASes and

proposed a customer cone metric which gives an indication on how global an AS is. We used this metric to seperate all ASNs into the three tiers.

- Customer cone $\geq$ 2000 $\rightarrow$ Tier 1 AS
- 2000 > Customer cone $\geq$ 200 $\rightarrow$ Tier 2 AS
- Customer cone < 200 $\rightarrow$ Tier 3 AS

We use the tier categorization in our visual interface to give more context to the BGP update path. The raw ASRank data is provided by CAIDA [CAIb, CAIa]. Because BGP updates only include AS numbers in the path, additional data is necessary to give more information about the owner of the AS. Therefore, we use the Routing Information Service (RIS) [RIP20] of RIPE NCC. The service provides detailed information about each AS and its prefixes. With those two additional data sources we enrich the BGP update data with more context information for the user.

3.2. User Groups

The analysis of interdomain routing and the internet infrastructure requires strong domain expertise. Thus, the targeted user of our application already has considerable domain knowledge. Our goal is to improve their analysis process by visualizing larger amounts of data and combining data sources to accelerate gaining insights. Therefore, our potential users are:

- Primary: AS Admins and Prefix Owners
- Secondary: AS, ISP, IXP Managers and Internet Regulators

Routing data is primarily analyzed to find anomalies or misconfigurations which lead to slower traffic or dropped packets. This is usually done by admins of ASes or the owners of IP prefixes to investigate issues or ensure a smooth service. These two user groups are our primary users. They are mainly interested in their own IP prefixes and prefixes of other large internet services that their customers have to reach consistently. Since it is possible to extract AS relations from the BGP update data, there is also an analysis interest from a business and regulation perspective. Therefore, we have a secondary user group, consisting of managers from ASes, Internet Service Providers (ISPs) and Internet Exchange Points (IXPs). Their focus is to overview the whole situation to see possible shortcomings and business opportunities. Our prototype is focussed on the primary user group and their tasks but also provides interesting views for the secondary user group.

3.3. Tasks and Requirements

We studied the related work for interests and tasks of our user groups. Additionally, we had an interview with a domain expert from an IXP before our development started. Later we were also able to get feedback on tasks from an AS admin and a researcher from RIPE NCC. The main task we found is to detect if an IP prefix has been hijacked by another AS. This means it was falsely announced by another AS, thus preventing traffic from reaching the services connected to those IP addresses. AS admins and IP prefix owners immediately need to know this as their services are directly impacted. Also they need to determine who caused this and which parts of the world are affected by this. Another task is to detect route leaks. Route leaks are falsely propagated routing announcements which were issued by an AS beyond their intended scope. This is for example the case if a Tier 3 AS starts to propagate routes which they got from their Tier 1 or 2 provider AS. This may overload the Tier 3 AS with traffic and lead to dropped traffic (a so-called black hole). For performance reasons it is also interesting to see which path the BGP update took geographically, as the routing policies are mostly guided by business contracts and not always optimized to take the shortest path. Therefore, one task is to analyze the peering between the ASes. For our secondary user group the performance perspective is very interesting as they are always looking for new business opportunities or optimization tasks. Based on that we formulated the following tasks that our prototype should support in a descending order of importance:

T1: Detect hijacks for specific IP prefixes
T2: Detect false propagation for a specific IP prefix
T3: Detect suspicious geographic detours of BGP update paths
T4: Approximate internal AS routing paths to make AS peering points visible

We derived the following requirements based on the data, user and task characterization:

R1: Make the large BGP update data accessible for analysis in a fast and progressive way
R2: Let the user filter the data to focus on relevant parts
R3: Highlight IP prefix hijacks so that they are immediately seen
R4: Visualize a geographic approximation of AS networks
R5: Show the BGP update path on a world map enriched with context data

Based on the explanation in Section 3.2 and 3.3, the requirements are connected to the tasks and users in the following way:

- Primary Users: **T1, T2, T3** $\rightarrow$ **R1, R2, R3, R5**
- Secondary Users: **T2, T3, T4** $\rightarrow$ **R1, R4, R5**

4. Approximation Algorithm

To achieve our goal of showing the geographical path of a BGP update over multiple ASes we need information about the entire underlying network. This entails the knowledge about the geolocations of IPs of any provider for any time span. We use GeoIP and AS-IP data from Maxmind as described in Section 3.1. Additionally, we require knowledge about the internal routing structure of every AS network. Both, the locations of data centers and the connections between them are pivotal for a good understanding about the routes that packets take over the Internet. Unfortunately, ground truth data for internal structures is scarce and thus we cannot rely on a lookup for each AS. Therefore, a good approximation of topologies of AS networks is necessary. Finally, knowledge about the actual routing over the AS network structure is crucial to predict reasonable routes. Since little information about intra-AS (iBGP) routing is published, this also has to be algorithmically approximated, based on reasonable constraints.

In this section, we illustrate our three approximation modules which are: Approximation for the data center locations, approximation of the connections between the data centers, and the approximation of the geographical routes based on the AS paths. For more details we provide the pseudo code in the supplemental material.

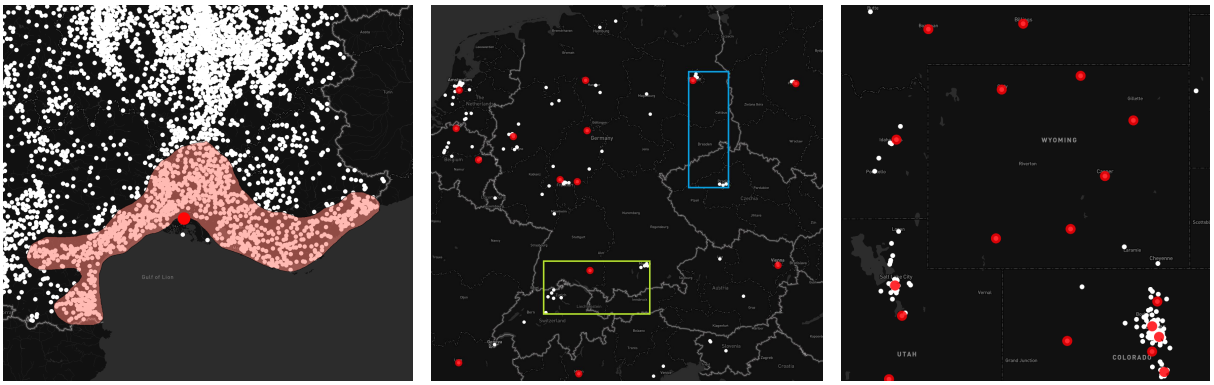

Figure 3: *Limitations of the tested data center approximation algorithms. White dots represent IP blocks from the GeoIP data set. Red dots represent estimated data centers.* ***Left:*** *DBSCAN approach provides clusters that can span over long distances, complicating reasonable approximation of the centroid.* ***Center:*** *k-d tree approach sometimes picks cells that contain multiple dense IP block areas. Picking a single centroid results in a bad approximation of data centers.* ***Right:*** *Multiple data centers in dense urban areas are not reasonable and should be merged to a single one. Picking each single IP block as a data center in sparse areas may lead to an overestimation for large AS networks.*

4.1. AS Data Center Approximation

To approximate the data centers of arbitrary ASes, we need to analyze the geo-positions of IP blocks automatically and to suggest reasonable locations for potential data centers. Since there is no comprehensive ground truth data that we could use as a training data set, we cannot use supervised approaches and machine learning techniques to facilitate the prediction of such potential data centers. To make unsupervised approaches work with GeoIP data, we analyzed properties of different ASes first, to find a common ground. This analysis and the insights gained from an interview with a domain expert led us to constraints for the analysis of IP block data and corresponding data centers. The first constraint is that for each AS, data centers are located closely to the majority of their managed IP blocks. The reason for this is that AS managers want to guarantee good reachability and connection speed for their customers. Another motivation is that AS managers prioritize their own AS network to lower their routing costs. The second constraint is that an AS does not have data centers in a geographically small area. Although this might not be true for large cities, our goal is to create an approximation on the city-level, thus we aggregate close data center candidates.

C1: an AS data center is in proximity of their managed IP blocks
C2: close by data centers are merged on a city level aggregation

Due to the different topologies of AS networks, an adaptive approximation had to be chosen that takes the constraints of data center locations into account. Our first ideas were to use DBSCAN or a k-d tree [Ben75]. But each technique by itself proved to be incapable to meet our constraints as shown in Figure 3. Constraint **C1** could not be satisfied in a stable way. Either the centroid is not in the proximity of most of the IP blocks, because of the lengthy shape of the cluster, or the centroid was geographically averaged in a k-d tree cell which resulted in a data center being in a sparse area. These limitations inspired us to implement a hybrid approach, combining k-d tree and DBSCAN. We perform the initial, coarse partitioning with the k-d tree algorithm. This way we rule out that the clusters can span over long distances and thus support constraint **C1**.

The fine clustering in the generated cells is performed by an adaptable DBSCAN implementation. This helps to identify a suitable number of clusters in the area based on the density distribution of IP blocks. Thus, we avoid that odd cluster centers can be chosen by our approximation algorithm. After comparison with the ground truth network maps of some ISPs we identified two areas for minor improvements. We chose the parameter estimation for the DBSCAN algorithm rather conservatively, to achieve nice positioning of the data centers. However, this resulted in an overconcentration of data centers in dense IP block areas. We can see this effect in Figure 3 on the right in Denver, Colorado. This compromises our constraint **C2**, to approximate data centers on the city level. In the example, we want to approximate only a single data center in Denver. To fix this issue without affecting the robust estimation of our implementation of the DBSCAN algorithm, we added a post-processing step. In this step, we merge data centers that are too close to each other. The location of the merged data center is determined by the largest IP block. The second minor issue can be seen on the right part of Figure 3 in Wyoming. There are multiple sparse IP blocks with only a few IP addresses. Since sparse and rather small IP blocks may qualify as data centers for small Tier 3 ASes, we handled this issue with an additional post-processing step. In this step, we categorize all data centers that manage only one single IP address as noise and filter them out. Further, for AS networks with a large amount of estimated data centers, we filter out the ones managing less than a certain threshold of IP addresses, if they are in the proximity of another data center that is above the threshold. Overall, the post-processing results in predictions that are more accurate for small and large AS networks.

4.2. Internal AS Connections

The data center approximation is only the first step to achieve our goal of predicting the routing through multiple ASes. A meaningful

approximation of the internal connections of data centers for each AS is a mandatory second step. As in the data center approximation case, we require an unsupervised approach to tackle this challenge due to the lack of comprehensive ground truth data. Based on the available network maps of some ISPs and an interview with a domain expert we identified constraints for our approximation model. We observed that connections are mostly present between data centers that are close to each other. Further, data centers that manage many IP blocks are often directly connected. Both features reflect that ISPs optimize their network based on speed and cost. Finally, we learned from the domain expert interview that, AS networks are not built like a minimal spanning tree, but have additional connections to reduce internal routing by avoiding detours along the minimal spanning tree.

C3: short connections between data centers are prioritized
C4: intercontinental connections are between large data centers
C5: data centers should reach others with only a few hops

At first we tried to use the NETHSIC algorithm [LSGB09]. NETHSIC is an unsupervised algorithm for structured network inference that uses kernels to model network topologies based on certain properties of the data. We chose *degree of connections* per data center as our target property. While the algorithm provided reasonable results, the time complexity $O(n^2)$ made it too slow for larger AS networks.

Therefore, we decided to use Kruskal's minimum-spanning-tree algorithm [Kru56] as the backbone of our implementation and extend it to fit our use case. On top of the spanning tree, we estimate new edges using the sorted edge data structure, already provided by Kruskal's algorithm. We traverse this edge list, which is sorted in ascending order, and check if the current edge is a suitable candidate for a connection. For this edge we calculate the shortest path over the existing spanning graph using the A* (A-Star) algorithm [HNR68]. This process automatically reduces the distances for other paths between data centers. Suitable edges are added to the graph iteratively resulting in networks that included extra edges (**C5**) and some edge crossings similar to the network map examples. The priorization of short connections (**C3**) and the priorization of connections between big data centers (**C4**) is balanced by the *distance metric*. The *distance metric* is a ratio of the *geographical distance* and the *sum of IP blocks*. We use the Haversine distance because it considers the earth's curvature. To mitigate the impact of inaccurate GeoIP a binning was necessary to prevent large IP clusters having an infinite impact on the connection priority.

4.3. Shortest Routing Path

The next step is to determine the geographical route from the next hop IP location to the prefix origin location through the AS path of the BGP update. Our approach creates a single graph that contains the next hop IP, the prefix origin, all AS networks from the AS path, and the connections between these entities. On this graph we can then calculate the shortest path using the A* algorithm.

For the construction of this graph, we need to complete three tasks. We need to connect the next hop to the first AS of the AS path. Then, we need to find reasonable connections between the ASes leading to the prefix origin location. Finally, we have to connect the origin AS with the prefix location. The obvious solution for the first and third task is to connect the next hop respectively the prefix location to the data center with the shortest distance. Thus, we generated two additional edges that complement the graph.

The second task poses a bigger challenge, since there are various ways to connect two different AS networks. After consulting a domain expert we specified the following constraints. ISPs have an inherent interest to route packets over their own network as far as possible before passing them to the next AS. This is simply most profitable for them. Another constraint is that ASes are locally connected, meaning that transitions between two ASes can only occur in locations where data centers from both ASes are in close proximity. Our last constraint is that routing should go in the direction of the prefix origin and prefer the shortest path without cycling and zigzag patterns.

C6: prioritize long paths in own AS network
C7: peered ASes are connected in one location
C8: prioritize shortest path in the direction of the target

Multiple data centers serve as transition point candidates between ASes based on their proximity to data centers of the next AS and the distance to the target location. This prevents the path from going through local optima which in turn might lead to an overall detour. After adding the potential transition edges to the graph we calculate the shortest path with A*. This satisfies all three constraints (**C6**, **C7**, **C8**) with one comprehensive solution. By opening up multiple options on how to traverse the graph between neighboring ASes, the A* algorithm provides several more optimal paths with short lengths and in the direction of the target prefix.

5. Visualization System

In this section, we describe the progressive data processing of the BGP update data and the storage of intermediate results created by our approximation algorithm. Both aspects are vital for ensuring a highly responsive visual analytics approach. Finally, the visualizations and interactions are explained.

5.1. Progressive Data Processing

BGP update data is large and split into small archives spread around all collectors. It takes time to download the data and process it. We developed a progressive data processing and visualization so that the user receives results incrementally as they are ready.

The process starts with receiving the query from the client containing an IP prefix, start time and end time. First, we call the CAIDA BGPStream broker to get metadata of all collectors that have data for this time frame and the URLs to the log files. Note that the BGPStream broker automatically puts in a limit of 30 minutes, so we have to call it multiple times if the query timeframe is larger. Then, we start multiple processes to download the data and store it for future queries. The downloaded log data includes all log entries, as we have not filtered by the queried IP prefix yet. After a file is downloaded, the BGPScanner application is called on the stored file with the IP prefix as a filter option. As described in Section 3.1 the log files are dumps in 5- or 15-minute intervals. The files

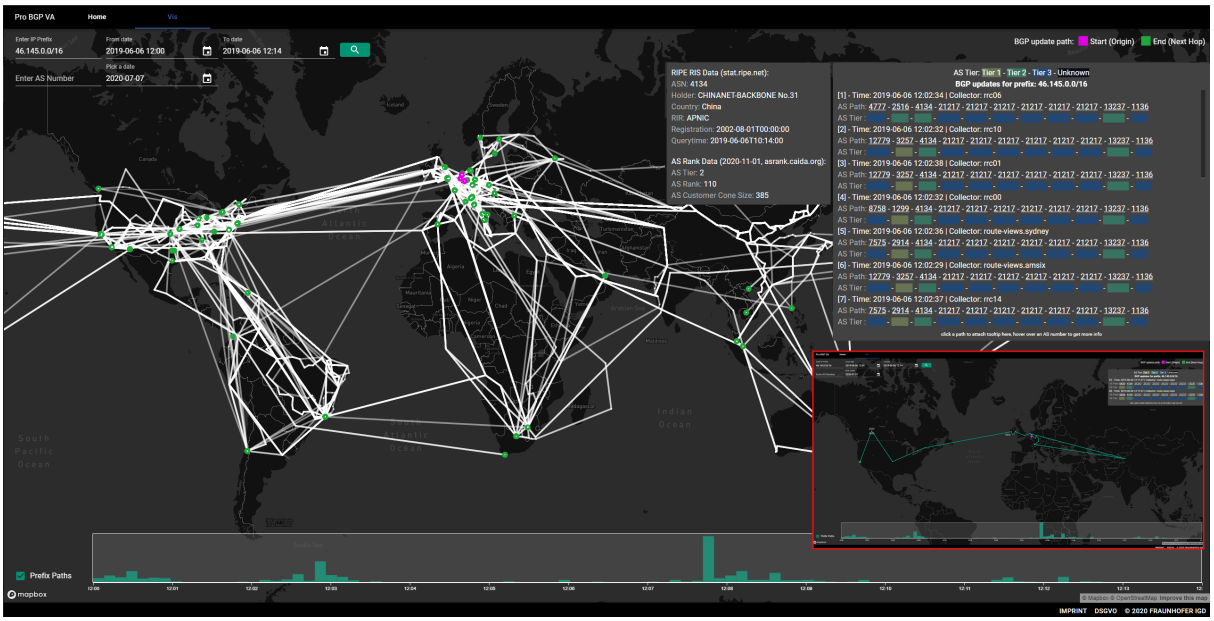

Figure 4: *Real world use case: Large European route leak from AS21217 sends traffic through China Telecom. Immediately, the large quantity of BGP updates (white paths) shows the global impact of this event. The color code on the right side shows how the routing hierachy was interrupted. The AS path shows that the Tier 3 AS21217 is in between two Tier 2 ASes, which in general should not be the case. In the bottom right, one path is highlighted by the user. It shows clearly how the path made a detour over China before being routed to the US.*

have very different sizes because of the position of the collector. Collectors at central ASes in Europe produce approximately three gigabytes of highly compressed data each day while collectors at smaller ASes in Africa, e.g., only produce around 150 MB. Thus, we can use a data-chunking approach to progressively return results as each part of the log dump is processed. Note that some of the log archives are corrupted. We abort these cases and return the paths found until the corrupted parts start. The resulting routing paths are then processed by our approximation algorithm using the closest GeoIP data available to the selected time frame. After generating a data center network for each AS in the update path we compute the shortest geographic path as described in Section 4.3. The geographic path is then streamed to the client to enable a progressively updating visualization. Finally, when all files were processed we send a *Finish* notification to the client and store the result in our database to speed up future queries. With this approach our service is always up-to-date as we allow the user to get the data directly from the collectors. For a large query of twelve hours for a random IP prefix our system processes on average 300 MB of highly compressed log files for each of the 54 collectors from Routeviews and RIPE. The first update path visualization is already available after 10 to 20 seconds and the finished visualization is ready after 8 minutes. For smaller time frames the first results are shown after a few seconds. With smart caching we provide results for queries with overlapping time frames in under one second. We are currently running this service on a smaller five year old server with 40 cores, 64GB RAM and HDDs. With this we provide a service for our defined user groups which satifies the requirement of a fast and progressive accessibility of BGP update data (**R1**).

5.2. Visualizations and Interactions

In this section we present our visual interface and show how it fullfils the proposed requirements from Section 3.3. We also refer the reader to the accompanying video that further highlights the visualization and interaction possibilities. The ProBGP visual interface depicted in Figure 1 shows our web service, accessible at https://probgp.igd.fraunhofer.de. On the start page, we introduce the topic and present an overview of the functions. After switching to the "Vis" tab, the map visualization is loaded and two rows of query inputs are visible on the top left.

The first row is used for the BGP update queries, where an IP prefix in CIDR notation can be entered (e.g. 146.140.0.0/16) and a start and end time. This reduces the very large data repository and helps the user focus on the relevant data (**R2**). Data can be queried from the beginning of the BGP recordings in 1999 up to now, with a limit of two days per query. After submitting a query, the loading spinner will show that the request is currently processed and single paths will be progressively added to the visualization as they arrive from the server. We tired different apporaches [AMSS19] to better reflect quality measures of the progressive computation but due to the unknown size of the real world data it was not possible to give a precise progress or stability feedback other than a progress bar which would take most of the time for the last few percent. The responses are not in a deterministic order because the server processes many files in parallel and each process of our approximation algorithm returns the results immediately. When a routing path is received, the path is rendered on the map with its next hop as a green endpoint and the origin of the prefix as a pink endpoint (**R5**). So if there is more than one pink endpoint that may be a multi-

origin prefix or a prefix hijack (**R3**). Since we provide an approximation of the routing path we looked at related work for uncertainty in geographic visualizations [ZB19] and experimented with visualizing the uncertainty of data centers and paths. This led to large overplotting problems and made it very difficult to understand the visualization. In fact, none of the geospatial visualizations in the recent survey [SYPB21] combined uncertainty visualization with an explicit geographic mapping. We also thought about using edge bundling [ZXYQ13] to have a clearer visualization. This would make BGP update paths difficult to follow through the countries, but it could be helpful for connections that cross oceans, which are in fact bundled. We decided to postpone this to future work as the partial application of edge bundling is challenging and provides less benefit than other features we developed.

At the bottom of the map a timeline histogram is updated with each new path to give an overview when the updates were recorded. A brush over the histogram gives the user the possibility to filter out paths to make the map visualization less overplotted if there are many paths over the whole timeframe (**R2**). This also affects progressively incoming data, maintaining the focus on the data relevant to the user. When hovering over a path the routing locations are highlighted and the corresponding AS numbers are shown so that the user can immediately see how far one AS network carried the BGP update. To view the raw data the user can hover over the paths on the map and select them. A tooltip window shows the raw data on the right side of the browser window, so that the view on the paths is not blocked. While hovering over a path, the other paths are hidden to have a clear view. When the user hovers over a path that is shared by many BGP updates, all raw data entries are displayed. As the list may get very long, the user can pin the tooltip by clicking on the path segment and then scroll inside the raw data tooltip. Inside the raw data tooltip the user can hover over an AS number to get more context as shown in Figure 4. The data is queried live from the RIPE RIS and shows the owner of the AS, the country, the registration time and the Regional Internet Registry (RIR) (**R3**). We also make use of the ASRank data set from CAIDA to categorize the ASes in tiers (**R5**). The legend at the top shows the color for each AS tier. Under each AS path the coloring shows the order of AS tiers the BGP update was propagated through. With this context it is possible to detect route leaks which we show in the use case in Section 6. With our visualization, it is possible to look at the BGP update paths geographically and find suspicious deviations from the shortest path.

The second query row is for the visualization of our AS data center approximation algorithm. First, the user can insert an AS number and pick a date to specify which GeoIP data is used. After submitting the query, the raw GeoIP data is shown with white dots with a low opacity, while the approximation algorithm is running. The data center aggregation usually takes less than 500ms so the data centers are almost immediately visible as colored dots, with a sequential multi-hue color map [HB03] overlayed at the top right. The raw data layer has a lower opacity and is displayed on top of the data center layer. This allows to see how the raw data was aggregated to the resulting data centers. Dark red dots depict very large data centers that have more than five million IP addresses in this area. Light yellow dots mean there are only one to a hundred IP addresses aggregated. Finally, the data center connections are displayed as white links between the data centers. This data might load a little bit slower as the algorithm has to test many edges (see Section 4.2). These loading times are only noticed on the first computation for an AS number on a specific date as the results are stored in our database. Finally, the user can show and hide layers at the bottom left by clicking the checkboxes. Our approximation algorithm enables managers from ISPs, ASes or IXPs to view internal AS networks (**R4**) which previously was not possible due to the lack of data. This enables them to find areas for optimization or new business opportunities.

6. Use Case

In the following, we demonstrate how the intended tasks can be performed with our prototype by following a real event in 2019. On June 6th 2019, AS21217 from Switzerland leaked over 70,000 routes to China Telecom AS4134. China Telecom did not filter this leak and propagated it to the global Internet. This caused a major amount of traffic that was destined for large European networks to be routed through the China Telecom network. We load the time frame from 12:00 to 12:14 UTC for the prefix 46.145.0.0/16 to look at a portion of the log data for this use case (see Figure 4). After the progressive computation is started, the first path is visible after about four seconds. The gist of the event is already visible after 30s when almost half of the data is processed. The computation ends after 1 minute and 30 seconds with 528 visualized paths. It is immediately apparent that there are many more routing paths in this time frame, when we compare it to Figure 1, for instance. The user can see the origins of the prefix marked as pink dots. If multiple pink dots are visible this indicates a possible prefix hijack (**T1**), but there are also multi-origin prefixes, so the user has to investigate further. The user can select each pink dot to see which AS has originated the prefix. In this case, AS1136 owns the prefix and originated it from all three locations. After hovering over some paths the user selects a set of paths and looks at the table to the right. It is immediately noticeable that AS21217, a Tier 3 AS, is in the middle of the AS path between the larger Tier 2 and Tier 1 ASes, which indicates a route leak or false propagation (**T2**). Going over the paths in detail (Figure 4 red box) the user can instantly see how the update path took a detour over China (**T3**). Further investigating the data table on the right, the user can hover over AS4134 pulling more information from the RIS which shows that the owner is Chinanet Backbone No.31. While hovering or selecting paths, the AS numbers are displayed at the routing nodes, which shows where the peering between ASes approximately was (**T4**). The use case shows that our prototype can handle real world data at a reasonable execution time and supports the intended tasks.

7. Evaluation

We had multiple evaluation stages with domain experts and were able to have experts from all our defined user groups for interviews at different stages of development. First, we had a pre-development interview with an expert from a large internet exchange point company. Second, we presented our pre-final work at the RIPE81 conference and were able to have follow-up interviews with a researcher from RIPE and an AS admin of a Swiss ISP. After the development we conducted an interview with the same expert from

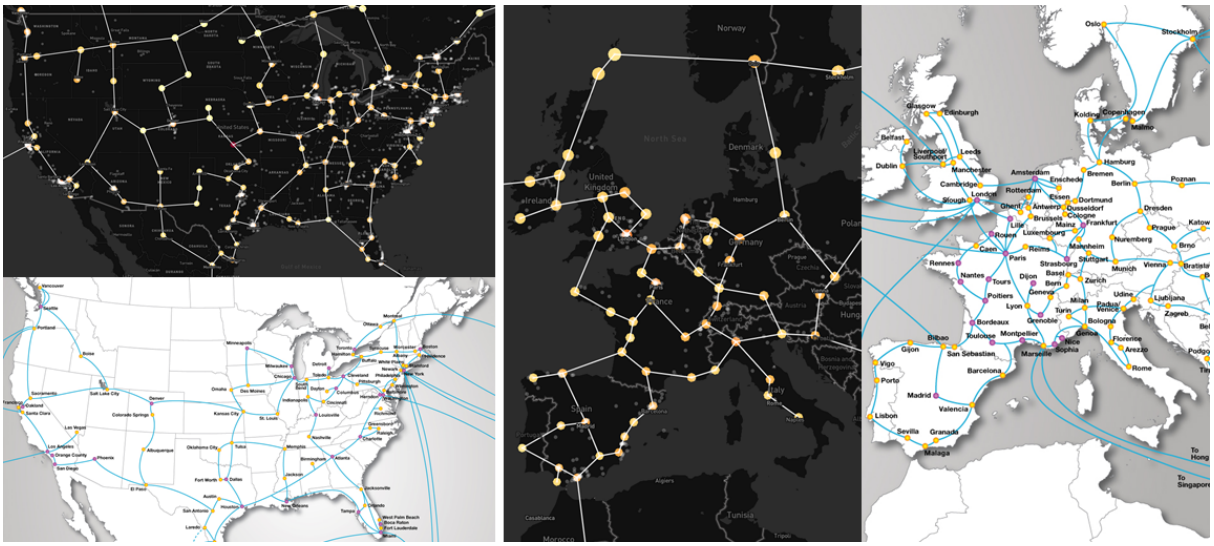

Figure 5: *Visual comparison to an infographic from Cogent AS174 [Cog20]. The infographic (white) does not show all data centers. Although our approximation algorithm produces fairly good results there are still balancing problems for our model constraints. On the left it is visible that our approximation generates more data centers in the USA than reported by the infographic. But on the right side our approximation generates less data centers in Europe than shown in the infographic.*

the IXP to verify our development progress and get additional feedback. Finally, we evaluated our approximation against the PeeringDB dataset and visually compared the internal network to official AS network maps.

7.1. Pre-Development Interview

Before starting the development of our approach we interviewed a domain expert from one of the largest internet exchange companies DE-CIX. We shared our ideas to implement a routing visualization with a higher geograpic accuracy than previous related works. The main problem we detected was the accuracy of the GeoIP data as our approximation can only be as good as the data we use. We created an estimation model which is not tailored for the GeoIP data, but will automatically perform better as the accuracy of the GeoIP data increases. We defined the constraints for our algorithm after consulting the domain expert to avoid vague assumptions. As a member of a large IXP the domain expert was more interested in the internal AS network than in the analysis of BGP updates. Based on the experts' feedback we implemented two separate query masks and visualizations, one for the analysis of BGP update paths and one for the analysis of our approximation results.

7.2. Mid-Development Feedback

After our first prototype was ready we were able to demonstrate it at the RIPE81 conference for internet infrastructure. There we explained our algorithm and showed the visual prototype. After the conference we were able to get in contact with one AS admin from Switzerland and a researcher from RIPE NCC. Their feedback was very different from the domain expert's comments in our first interview. They were much more interested in the BGP update visualization and gave multiple suggestions on how to boost the geographic accuracy. They also suggested to have multiple views on the data to close the gap between an approximated view and a non-approximated visualization of the raw data. They also suggested to include ASRank data to show the "Pyramid" structure of an AS path. The pyramid symbolizes the start of the AS path at a Tier 3 AS going over Tier 2 and Tier 1 ASes back to a Tier 3 AS. This makes it possible to see route leaks if there are multiple switches of AS tiers in the middle of the AS path

7.3. Final Evaluation Interview

In our final evaluation we were able to interview the domain expert from DE-CIX again. We showed him our visual comparison that is described in the next section. He stated that it is hard to compare against an infographic of an AS as they may not show all their data centers. But as our approximation is based on the location of their managed IP addresses, we might find more than they show on their map. The expert also stated that the accuracy of the Maxmind GeoIP database might not be optimal for estimating routers, as it is rather focused on end-user geo-locations. He recommended using the PeeringDB database where in the past years more ISPs started to provide accurate locations of their data centers. We evaluated against this dataset (see Section 7.4) and are currently looking into incorporating this data source into our algorithm to optimize the geographic approximation of data centers.

7.4. Quantitative Evaluation and Visual Comparison

Ground truth data for AS data centers and their internal connections is hard to find. PeeringDB.com [Pee21] provides a user-managed database with AS data center locations. However, the database does

not provide complete datasets for all ASes, because some ASes do not want to share this information. We used the available addresses for all Tier 1 and Tier 2 ASes and determined the geographical coordinates with forward geocoding. Then, we computed precision and recall while counting a true positive if our estimation is within a 90km radius of the ground truth location using the Haversine distance metric. It is the same radius we use to merge data centers in our algorithm to have an approximation on the city-level. For the top 10 Tier 1 AS of CAIDA ASRank we reach a precision of 73% and a recall of 78%. But we had to exclude two Level3 ASes as there was no ground truth data. For all Tier 1 and Tier 2 ASes we achieve a precision of 38% and a recall of 65%. The precision suffers strongly from incomplete ground truth data, as there are several large ASes, for which we estimate hundreds of locations but PeeringDB only has one or two locations registered. It is not clear if the data is incomplete or if we overestimated, so we did rather not want to filter out these cases. For the AS174 by Cogent, which we use in the following we achieved a precision of 84% and a recall of 93%.

The evaluation of the internal AS network was more difficult because there is no quantitative ground truth data. Only a few of the top tier ASes provide a network map with the internal connections like Hurricane Electric [Hur20], Cogent [Cog20] and CenturyLink (Level3) [Cen20]. We took the network maps of multiple ASes with the most IPv4 peers worldwide and used them during our development. Here, we show a comparison with the fourth-largest AS according to CAIDA ASRank (01/2021): AS174 by Cogent. Their network consists of one main AS that we can more easily compare with, while others like CenturyLink have multiple AS numbers in their network. Looking at the topology of the network, there are some important differences, which make the general approximation more difficult. Thus, for future improvements it would be very helpful to gather a ground truth data set with information on the connections of multiple AS networks to also perform a quantitative evaluation on the internal connections.

Looking at the network map of Cogent (Figure 5, white map) we identify that they maintain many data centers in the US and Europe. Our algorithm generates a slightly higher data center density in the US (Figure 5, dark map). There, our approximation predicts some more small data centers especially in the northern part of the country. In southern Europe, we see similar data center densities with slight shifts in their locations, but in central Europe our algorithm is aggregating too much. This shows that it is difficult to balance the estimation algorithm between dense and sparse regions. Overall, the approximated connection density and structure of the internal network is close to the original. Noticeable is that connections over long distances (e.g. Australia to South America) which go through oceans are generated by our algorithm, but are not present in the real world. Therefore, we are thinking of integrating penalties on connections over certain regions as there are only some areas with underwater cables. Currently, these connections are estimated based on their geographical distance, disregarding economic factors.

8. Discussion and Future Work

During our development, we faced some challenges and limitations and we noticed some extensions that could improve our approach. One big challenge was the processing of the large BGP update data. BGP update queries for long time periods over all collectors require large downloads and much computation power to achieve a responsive web service. Further, the data access does not allow to query for a specific location. Therefore, it is not possible to only get data for a specific geographical region without downloading the data of all collectors. Also, the accessible time frames are discretized in 5 minute (RIPE) or 15 minute (Routeviews) intervals. This limits the amount of the progressive features we could implement. Acquiring a better accessibility to the BGP update data that allows for more specific queries would unlock progressive capabilities we are highly interested in, like steering the progression and the monitoring of progressive quality indicators [AMSS19] in a specific geographical region. The second big challenge was the modeling of our algorithms, as there is only limited ground truth data to compare against. Looking at the different AS network maps showed us that companies have very different network topologies. Real network connections are not always optimized for the shortest geographical path but are also driven by economic, political or environmental requirements. The main limitations of our approach are the accuracy of the GeoIP data, which directly influences our approximation accuracy, and the fact that there is only incomplete ground truth data we could evaluate against.

Next, we will integrate all the valueable suggestions by the different user groups we were able to gather. We are going to improve our algorithm by modeling in factors like costs for connections going over land or water, as this covers environmental and political factors. The addition of all BGP log collectors on the map is also planned in next iterations, as this will add to the comprehension where the log data is from. A future goal is to integrate automatic suggestions for suspicious routing paths to guide the user to interesting points in the data.

9. Conclusion

In this paper, we presented a novel geographic AS data center network approximation algorithm and an always up-to-date progressive visual analytics application to analyze BGP updates. We summarized the related work in network visualization and progressive visual analytics and highlighted the connections to our apporach. We defined two user groups, their tasks and made use of GeoIP, AS-IP, ASRank data and BGP update logs in our system. We proposed a novel geographic approximation algorithm for AS data centers and an algorithm to estimate the internal connections between the data centers. We created a progressive visual analysis system for BGP updates and described our interaction design in detail. We showed how our approach can be used to accomplish the tasks with a real world use case. Finally, we reported on multiple interviews during our development and evaluated our approximation quantitatively and qualitatively.

10. Acknowledgments

This research work has been funded by the German Ministry of Education and Research and the Hessian State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[AMSS19] ANGELINI M., MAY T., SANTUCCI G., SCHULZ H.-J.: On quality indicators for progressive visual analytics. In *EuroVA@ EuroVis* (2019), pp. 25–29. 7, 10

[ASSS18] ANGELINI M., SANTUCCI G., SCHUMANN H., SCHULZ H.-J.: A review and characterization of progressive visual analytics. In *Informatics* (2018), vol. 5, Multidisciplinary Digital Publishing Institute, p. 31. 3

[BEF17] BADAM S. K., ELMQVIST N., FEKETE J.-D.: Steering the craft: Ui elements and visualizations for supporting progressive visual analytics. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 491–502. 3

[Ben75] BENTLEY J. L.: Multidimensional binary search trees used for associative searching. *Communications of the ACM 18*, 9 (1975), 509–517. 5

[BEW95] BECKER R., EICK S. G., WILKS A. R.: Visualizing network data. *IEEE Trans. Vis. Comput. Graph. 1* (1995), 16–28. 2

[CAIa] CAIDA: As relationships dataset, <2020-11-01>. . URL: https://www.caida.org/data/as-relationships/. 4

[CAIb] CAIDA: Caida as rank. . URL: http://as-rank.caida.org/. 4

[Cam21] CAMBRIDGE INTELLIGENCE:. https://cambridge-intelligence.com/use-cases/cybersecurity/, 2021. URL: https://cambridge-intelligence.com/use-cases/cybersecurity/. 2

[Cen20] CENTURYLINK: CenturyLink Global Network, multiple AS numbers. Accessed on 07.07.2020, 2020. URL: https://www.centurylink.com/asset/business/enterprise/network-map/centurylink-network-maps.pdf. 10

[Cog20] COGENT:. Accessed on 07.07.2020, 2020. URL: https://www.cogentco.com/en/network/network-map. 9, 10

[DBMPP03] DI BATTISTA G., MARIANI F., PATRIGNANI M., PIZZONIA M.: Bgplay: A system for visualizing the interdomain routing evolution. In *International Symposium on Graph Drawing* (2003), Springer, pp. 295–306. 2

[FFNS19] FEKETE J.-D., FISHER D., NANDI A., SEDLMAIR M.: Progressive data analysis and visualization (dagstuhl seminar 18411). In *Dagstuhl Reports* (2019), vol. 8, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 3

[FFV*12] FISCHER F., FUCHS J., VERVIER P.-A., MANSMANN F., THONNARD O.: Vistracer: a visual analytics tool to investigate routing anomalies in traceroutes. In *Proceedings of the ninth international symposium on visualization for cyber security* (2012), pp. 80–87. 2

[FP16] FEKETE J.-D., PRIMET R.: Progressive analytics: A computation paradigm for exploratory data analysis. *arXiv preprint arXiv:1607.05162* (2016). 3

[FPD*12] FISHER D., POPOV I., DRUCKER S., ET AL.: Trust me, i'm partially right: incremental visualization lets analysts explore large datasets faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 1673–1682. 3

[GSH*17] GHARAIBEH M., SHAH A., HUFFAKER B., ZHANG H., ENSAFI R., PAPADOPOULOS C.: A look at router geolocation in public and commercial databases. In *Proceedings of the 2017 Internet Measurement Conference* (2017), pp. 463–469. 3

[HB03] HARROWER M., BREWER C. A.: Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37. 8

[HB05] HEER J., BOYD D.: Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (2005), pp. 32–39. doi:10.1109/INFVIS.2005.1532126. 2

[HNR68] HART P. E., NILSSON N. J., RAPHAEL B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics 4*, 2 (1968), 100–107. 6

[Hur20] HURRICANE ELECTRIC: AS6939 Hurricane Electric IP Transit Network. Accessed on 07.07.2020, 2020. URL: https://he.net/about_network.html. 10

[KKEM10] KEIM D., KOHLHAMMER J., ELLIS G., MANSMANN F. (Eds.): *Mastering the information age : solving problems with visual analytics*. Goslar : Eurographics Association, 2010. URL: https://diglib.eg.org/handle/10.2312/14803. 2

[Kru56] KRUSKAL J. B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society 7*, 1 (1956), 48–50. 6

[KVR17] KOMOSNY D., VOZŇÁK M., REHMAN S. U.: Location accuracy of commercial ip address geolocation databases. 3

[LAA*20] LIU F., ANDRIENKO G., ANDRIENKO N., CHEN S., JANSSENS D., WETS G., THEODORIDIS Y.: Citywide traffic analysis based on the combination of visual and analytic approaches. *Journal of Geovisualization and Spatial Analysis 4* (12 2020). doi:10.1007/s41651-020-00057-4. 2

[LHD*13] LUCKIE M., HUFFAKER B., DHAMDHERE A., GIOTSAS V., CLAFFY K.: As relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), pp. 243–256. 3

[LSGB09] LIPPERT C., STEGLE O., GHAHRAMANI Z., BORGWARDT K.: A kernel method for unsupervised structured network inference. In *Artificial Intelligence and Statistics* (2009), pp. 368–375. 6

[Pee21] PEERINGDB:. https://www.peeringdb.com/, 2021. URL: https://www.peeringdb.com/. 9

[PLvdM*16] PEZZOTTI N., LELIEVELDT B. P., VAN DER MAATEN L., HÖLLT T., EISEMANN E., VILANOVA A.: Approximated and user steerable tsne for progressive visual analytics. *IEEE transactions on visualization and computer graphics 23*, 7 (2016), 1739–1752. 3

[PMT13] PAPADOPOULOS S., MOUSTAKAS K., TZOVARAS D.: Bgpviewer: Using graph representations to explore bgp routing changes. In *2013 18th International Conference on Digital Signal Processing (DSP)* (2013), IEEE, pp. 1–6. 2

[RIP20] RIPE NCC: RIPE Routing Information Service, 2020. URL: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris. 4

[RS09] ROSENBAUM R., SCHUMANN H.: Progressive refinement: more than a means to overcome limited bandwidth. In *Visualization and Data Analysis 2009* (2009), vol. 7243, International Society for Optics and Photonics, p. 72430I. 3

[SDB16] SYAMKUMAR M., DURAIRAJAN R., BARFORD P.: Bigfoot: A geo-based visualization methodology for detecting bgp threats. In *2016 IEEE Symposium on Visualization for Cyber Security (VizSec)* (2016), IEEE, pp. 1–8. 2

[SPG14] STOLPER C. D., PERER A., GOTZ D.: Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1653–1662. 3

[SYPB21] SCHÖTTLER S., YANG Y., PFISTER H., BACH B.: Visualizing and interacting with geospatial networks: A survey and design space, 2021. arXiv:2101.06322.

[Tho20a] THOUSANDEYES:. Accessed on 07.07.2020, 2020. URL: https://datacentrenews.eu/story/top-internet-outages-of-2019-thousandeyes. 2

[Tho20b] THOUSANDEYES:. Accessed on 07.07.2020, 2020. URL: https://www.thousandeyes.com/outages. 2

[USSK18] ULMER A., SCHUFRIN M., SESSLER D., KOHLHAMMER J.: Visual-interactive identification of anomalous ip-block behavior using geo-ip data. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)* (2018), IEEE, pp. 1–8. 3

[ZB19] ZOU L., BROOKS S.: A dynamic approach for presenting local and global information in geospatial network visualizations. *GeoInformatica 23*, 4 (2019), 733–757. 8

[ZXYQ13] ZHOU H., XU P., YUAN X., QU H.: Edge bundling in information visualization. *Tsinghua Science and Technology 18*, 2 (2013), 145–156. 8