

University of Glasgow | School of Computing Science

A Map of The Internet

Aleksi Peltonen

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 21, 2017

Abstract

The Internet is an extremely complex network that keeps on growing from year to year. It consists of more than 80 000 subnets, called autonomous systems (ASs). Each AS is a collection of devices that are connected and aware of each other. The aim of this project is to visualize the size and shape of the Internet and to collect and analyse statistics about the two IP versions in use. The results consist of visualizations from the past 20 years and an overview of how the network has changed.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

1 Introduction **1**

1.1 Background and motivation 3

1.2 Project description 3

2 Background information **5**

2.1 Internet topology and routing 5

2.2 Border Gateway Protocol (BGP) 5

2.3 Data sources and tools 6

3 Project requirements **7**

4 Implementation **8**

4.1 Summary 8

4.2 Collecting data 8

4.3 Presenting the results 9

4.3.1 Drawing maps 9

4.3.2 Collecting statistics 13

5 Results **15**

5.1 Statistics 15

5.2 Maps 21

5.2.1 Recent data 21

5.2.2 Historical data 22

6 Development, conclusions and future work **28**

6.1 Developing software for the project 28

6.2 Conclusions and future work 29

Appendices **31**

A Running the Program **32**

Chapter 1

Introduction

The Internet is a network of networks that communicate with each other. Each individual network, called an autonomous system (AS), is maintained by a group of administrators and consists of routers, switches, servers and other devices that are connected to each other. The size of an AS varies from small companies at the edge of the network, to large Internet service providers (ISPs) connecting hundreds of other autonomous systems to the network. Together all these individual subnets form a worldwide network known as the Internet. [10]

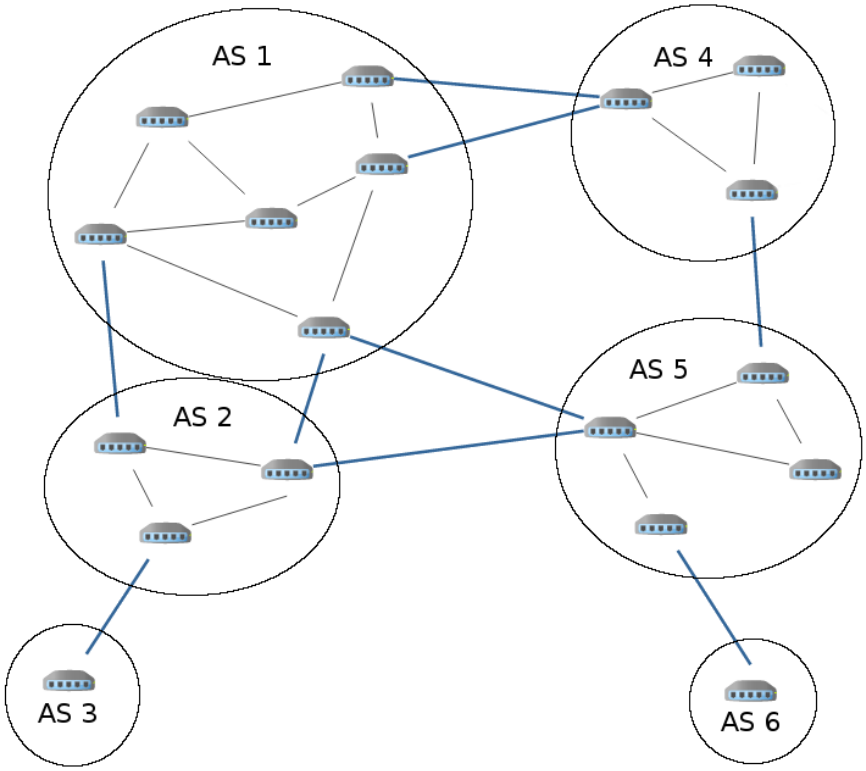


Figure 1.1: The structure of a network consisting of autonomous systems.

There are different ways of understanding the general structure and basic functionality of the Internet. One way of looking at it, is by considering it as a collection of different types of networks that can be (loosely) categorized according to their size and functionality. The three main categories are:

- **tier 1 networks** (or global networks) that compose the core of the Internet. There are currently around a dozen tier 1 networks in the Internet. Their task is to forward packets settlement free to each other. By connecting to one or more of the tier 1 networks, smaller subnets can reach the majority of the Internet without needing a large amount of connections.
- **tier 2 networks** (or regional networks) that participate in the settlement free peering, but also purchase or sell peering from/to other networks. These are known as transit networks.
- **tier 3 networks** (or local networks) that operate exclusively by buying peering from higher category networks.

In addition to these three network types, the Internet includes several hundred Internet Exchange Points (IXPs) that are used to connect ISPs to each other. However, these definitions are only indicative and are mainly used to understand the structure of the Internet, rather than being strict rules. [10]

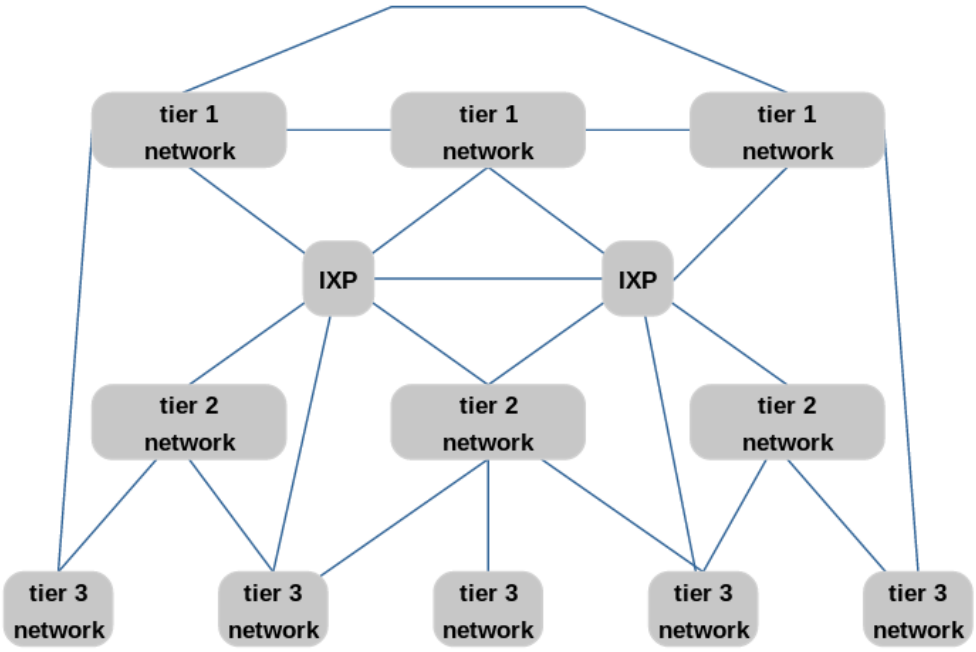


Figure 1.2: A simplified illustration of the connections between different network types.

Due to the fundamental principles of how the Internet is structured, it does not consist of a set group of networks, but rather a changing amount subnets joining, leaving, and constantly changing in size. As a result, collecting information about the network as a whole requires combining data from different sources and collectors to ensure a sufficient accuracy of the results. The objective of this project is to collect, combine and summarize data representing the entire network, its structure and development throughout the years. The end result will be a set of visualizations and graphs representing snapshots of how the Internet looked and behaved in different points in time. This report will introduce the necessary background information for the project, describe the development process of the software and present the main results. No preliminary information about the structure or functionality of the Internet is required from the reader, as it will be presented at the beginning of this report. However, basic programming knowledge (mainly C and Python) will help in understanding the discussion of how the software was implemented.

1.1 Background and motivation

The development of the Internet began in the 1960s and it became commercially available in the late 1980s. Since then, it has been growing extremely fast as a result of not only new users, but also new technology, such as mobile devices, smart-TVs and the Internet of things, giving each user a reason to connect multiple devices to the Internet simultaneously. Each new device that is connected needs to be allocated an IP (Internet Protocol) address for the duration of the connection. There are currently two versions that are being used: IP version 4 (IPv4) and version 6 (IPv6). IPv4 was first introduced in 1981, when the Internet was still in its early stages. [9] The IPv4 address pool is 32 bits and it contains approximately 4.3 billion (2^{32}) addresses, an amount that at the time was considered to be enough for the network, even in the future. IP addresses are being allocated in 'blocks' of addresses, where the size of each block determines how many addresses the owner is entitled to use. [8] The allocation of addresses is coordinated by the Internet Assigned Numbers Authority (IANA). Blocks of addresses are assigned in a hierarchical manner to Regional Internet Registries (RIRs), who in turn assign them to Internet service providers. The ISPs are responsible for allocating addresses to each device that gets connected to their network. [6] The last blocks of IPv4 addresses were assigned by IANA in February 2011. [5] IPv6 deployment began in 1999 as a successor to IPv4. [6] The IPv6 address pool is 128 bits and it contains approximately 3.4×10^{38} (2^{128}) addresses. Although the amount of addresses is enormous and many times greater than its predecessors, it *could* theoretically be exhausted as a result of careless allocation of huge blocks. As the last blocks of IPv4 addresses have now been allocated, the use of IPv6 addresses can be expected to grow radically. However, the amount of allocated addresses in blocks does not necessarily correspond to the amount of addresses that are actually being used and advertised.

The Center for Applied Internet Data Analysis (CAIDA) is a collaboration of organizations conducting research on the structure of the Internet, its behaviour, usage and evolution. One of their main contributions is mapping and analysing the Internet. [1] Since the beginning of the year 2000, CAIDA has been collecting data from various BGP collectors and creating visualizations of the Internet topology at a macroscopic scale. The maps illustrate the autonomous systems and their connections geographically. [2] This project aims to replicate the mapping tool developed by CAIDA to verify their results and to analyse the growth of the amount of IPv6 addresses being advertised.

1.2 Project description

The purpose of this project is to develop an application for analysing and visualizing the Internet and its evolution over time. The result will be a software that downloads and combines data about registered autonomous systems, the prefixes they advertise and the connections they have to other systems. This information is used to create visualizations and collect statistics that can be used to study the changes of the network. The project can roughly be divided into two parts: (1) software implementation and visualizations; and (2) data collecting and statistics. Part one, discussed in chapter 4, will mainly consist of programming and testing, while part two, presented in chapter 5, will consist of analysing the results. The second part of the project will focus on analysing differences and changes in the two currently used IP-versions. Questions that are of particular are interest are:

- How much has the Internet changed over time and how fast are the changes occurring?
- How many of the allocated IPv4 addresses are actually being used?
- How many prefixes does each AS advertise on average?
- How many neighbours does each AS have on average?
- How fast is the use of IPv6 growing? Does it correspond to the growth of IPv4 in its earlier stages?

The goal is to collect and combine data from a number of independent data collectors and present the result as maps illustrating the macroscopic topology of the Internet. Data will be collected from the past 17 years (2000-2017) for IPv4, and from the past 13 years (2004-2017) for IPv6. Each visualization will illustrate the Internet as it looked during the first three days of the month. Furthermore, statistics will be collected and analysed from the first month of each year and presented as bar charts showing the changes in the network over time.

The remainder of this report is structured as follows. Chapter 2 presents necessary background information about routing, the topology of the Internet and the Border Gateway Protocol (BGP). It also briefly describes data sources and tools used for the project. Chapter 3 briefly explains what the requirements for the software are and what the end user should be able to do with it. Chapter 4 explains the implementation of the software and describes what statistics are being collected and how the results are presented. Chapter 5 presents the results of the project. The most relevant visualizations and graphs are displayed and discussed in detail. Chapter 6 gives a brief summary of the results, reflects on the implementation part of the project and discusses future work.

Chapter 2

Background information

2.1 Internet topology and routing

Understanding how packets are transferred from one node in a network to another requires some knowledge about the structure and the topology the network. The first thing to understand is the difference between forwarding and routing. As the name suggests, forwarding involves transmitting a packet from an incoming link to an outgoing link within a router. On the contrary, routing involves all necessary steps for finding the (optimal) route between two nodes. The basic idea of routing in the Internet is based on forwarding tables in routers. When a host wants to sends a packet to a router, it examines the destination and decides the next step based on the best matching value in its current routing table. In order to accurately send packets to the correct destination, the routing tables are constantly updated as the network changes. [10]

The size of the Internet makes it impossible for each router to maintain a list of the entire network. Hierarchical routing is used to avoid problems with scalability and administrative autonomy. Instead of having each individual router function on its own, they are divided into groups called autonomous systems. Each AS has a unique 16 or 32 bit autonomous system number (ASN) and is maintained by a set group of administrators. The routers of an AS all use the same intra-autonomous system routing protocol to communicate with each other, and they are all aware of the other routers belonging to the same system. [10] There are currently more than 80 000 registered autonomous systems. [3] The allocation of AS numbers is done similarly to IP addresses: IANA allocates numbers as blocks to RIRs who assign them to network operators. [4] At the edge of each AS is one or more gateway routers that are responsible for communicating with other ASs. This allows each AS to use their preferred routing protocol for internal communication, as long as the gateway routers for all ASs in the network use the same inter-AS routing protocol to move packets between systems. The protocol used in the Internet is called Border Gateway Protocol 4 (BGP4). [10]

2.2 Border Gateway Protocol (BGP)

Border Gateway Protocol is the standardized inter-AS routing protocol of the Internet. Its four main purposes are: (1) maintaining information about reachability between neighbouring ASs; (2) distributing reachability information inside the AS; (3) deciding the best routes between subnets; and (4) allowing ASs to advertise their existence. The relevant features for this project are the first and the last one, as they are essential to maintaining a list of advertised IP addresses and connections between neighbouring ASs. BGP works over a semi-permanent TCP connection between two routers called BGP peers. The connection between two routers is called a BGP session, and it is either internal (iBGP) if both routers belong to the same AS, or external (eBGP) if they are edge

routers from different ASs. These sessions are used to exchange information about which subnets (expressed as prefixes) can be reached through neighbouring ASs. Thus, placing routers in central locations of the Internet can be used to create snapshots of the Internet by storing the routing tables as BGP dumps from a given point of time. [10]

2.3 Data sources and tools

Drawing a topology map of autonomous systems and collecting statistics requires three types of data:

- **AS numbers** and their corresponding organisations. BGP tables use AS numbers to distinguish between different ASs. An updated list of registered autonomous systems is downloaded from [3] at the beginning of each data collection.
- **IP geolocations**. The maps are structured based on the average longitudes of the prefixes each AS advertises (see chapter 4 for further details). An updated list of geolocations is downloaded from [7] at the beginning of each data collection.
- **BGP data**. BGP dumps are collections of BGP tables from a specific time period. These tables contain information about which prefixes belonged to different autonomous systems, how they were connected to each other and which the optimal routes from one AS to another was at the moment. To ensure sufficiently accurate results, data is collected and combined from the first three days of each month and from multiple data collectors. IPv4 data is collected from the University of Oregon Route Views Project[1] (collector: route-views2) and the RIPE NCC Routing Information Service (RIS)[2] (collectors: rrc00, rrc01, rrc02) and IPv6 data from the University of Oregon Route Views Project (collector: route-views6).

The most important tools used for collecting and visualizing data:

- **bgpdump**[3] and **BGPStream**[4]. The initial version of the software used bgpdump, a binary tool developed by RIPE NCC, to parse BGP data. The tool requires the user to download compressed data collections, which it parses into human-readable form. In November 2016 Orsini et al. [11] published a paper introducing BGPStream, a tool similar to bgpdump. The advantage of BGPStream over bgpdump is that it downloads data at runtime. This tool was used to replace bgpdump, as it removed the unnecessary step of downloading and removing data collections (see chapter 6 for further details).
- **matplot.pyplot**[5] and **networkx**[6] are used to visualize the data. The maps consist of nodes representing autonomous systems and edges representing the connections between different networks. Data is collected from previously created files and stored as A4 sized PDF files.

[1] `http://www.routeviews.org/`
[2] `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`
[3] `https://bitbucket.org/ripencc/bgpdump/wiki/Home`
[4] `https://bgpstream.caida.org/`
[5] `http://matplotlib.org/api/pyplot_api.html`
[6] `https://networkx.github.io/`

Chapter 3

Project requirements

An important part of this project is to develop a software for combining data and using it for creating visualizations and collecting statistics. The software will be used for analysing the changes of IPv4 and IPv6 during the past two decades. After all data has been collected, the software becomes obsolete for the sake of the project. However, as new data becomes available, it can be used to improve the results and conduct further research on the evolution of the Internet. Thus, an important part of the implementation phase is to ensure the suitability of the software for future use.

Requirements for the software:

- **Modularity**. The software should consist of independent parts that can be used for different tasks (such as combining data, drawing maps, collect statistics etc.), allowing the users to enable and disable individual parts for each run. The reason for this is to avoid the unnecessary overhead of running parts of the software that might not be needed for a particular purpose.
- **Error handling**. The main data sources are BGP tables consisting of data from all over the Internet. A single wrongly configured router may cause irregularities in the data, such as faulty AS numbers or prefixes. In case of damaged data, the execution should continue normally and the user should be alerted about the error that occurred. Furthermore, in case the machine running the software crashes or reboots, the amount of lost data should be as small as possible.
- **Continuous feedback**. Collecting data from a longer period of time might take days or even weeks to complete. The users should continuously be informed about the current state of the execution.
- **Simplicity**. The software should be reasonably easy to understand and modify in case the input data changes (e.g. new BGP table format) or new features needs to be implemented.

A user should be able to:

- **Define an arbitrary time interval for data collection**, as long as there is data available from the collectors. The entire process is coordinated by a shell script (see appendix A for details) that allows the user to specify the start and end date for collecting data.
- **Easily enable and disable parts of the software**. For example, a user might be interested in collecting statistics, but not in drawing maps or creating animations.

Chapter 4

Implementation

4.1 Summary

The program consists of three main parts that are executed consecutively for each map:

1. Combining AS names and numbers with BGP data. The purpose of the first part is to iterate through BGP dumps and combine the collected data with corresponding AS data. This part is written in C.
2. Complementing the collected data with geolocations. The second part adds the approximate longitude of each AS to be used for the visualization. This part is written in Python.
3. Drawing the map. Finally, a map is drawn using the information collected in steps 1 and 2. This part is written in Python.

Two additional features of the program are collecting statistics and creating animations. However, these parts are only executed once all maps are already created.

4.2 Collecting data

Parsing BGP data

BGP data is collected using `BGPStream` and parsed with a custom C program written for the project. The program starts by collecting AS numbers and their owners from [3] and stores them in structs representing an autonomous systems.

```
typedef struct AS {
    int num; // AS number
    char *name; // AS owner
    int degree; // Number of neighbours
    struct Prefix *prefixes; // Advertised prefixes
    struct AS *next;
    struct Neighbour *neighbours;
} AS;
```

Once all the registered ASs are stored in memory, a `BGPStream` pipe is opened and used to retrieve BGP data line by line. Each line contains a prefix and information about its source and the optimal AS route to reach it. Unique prefixes and neighbours are stored in the corresponding AS struct. When all data is parsed, the structs are stored in a CSV (comma-separated values) file where each line represents an AS.

AS	NAME	NEIGHBOURS	PREFIXES	LONGITUDE	DEGREE
0	-Reserved AS-, ZZ	38266 50459 15447 8530 48072 6849 ...	1.38.0.0/17 31.148.221.0/24 31.169.58.0/24 ...	-2.751510715596328	100
1	LVLT-1 - Level 3 Communications, Inc., US	11537 13227 26114 63264 10753 21976	4.31.236.64/29 4.34.12.0/24 4.34.13.0/24 ...	-95.32355882678472	6
2	UDEL-DCN - University of Delaware, US	197592 134006 134806 58818 56124 9829 ...	91.225.188.0/22 103.47.153.0/24 103.75.56.0/24 ...	49.03379389952151	15
3	MIT-GATEWAYS - Massachusetts Institute of Technology, US	174 3356 23248 134056 10578 7015 293 ...	38.97.116.0/24 45.117.29.0/24 45.117.31.0/24 ...	12.281135312831386	24
4	ISI-AS - University of Southern California, US	30982 226 264023 63483 567 63264 262906	80.248.77.0/24 128.9.0.0/16 143.0.252.0/22 ...	-90.33847971014492	7
5	SYMBOLICS - Symbolics, Inc., US	58967 29898	59.152.40.0/24 59.152.41.0/24 59.152.42.0/24...	22.341790140845063	2
6	BULL-HN - Bull HN Information Systems Inc., US	23682 16128 3356 1239 6 262324	101.0.52.0/24 129.185.24.0/24 129.185.30.0/24 ...	-42.126926916221024	6
7	UK Defence Research Agency, GB	7961786	66.181.128.0/20 146.80.0.0/16 192.5.28.0/24 ...	-8.707761073825509	2
8	RICE-AS - Rice University, US	135737 3356 8 4557 2914	103.66.236.0/22 128.42.0.0/16 128.42.210.0/24 ...	-64.57966504065041	5

Table 4.1: The first few lines from a sample CSV file (January 2017).

Parsing GeoLite data

Visualizing the collected data requires the locations of each AS. Locations are obtained from [7] and parsed with a Python script. Each line contains a subnet expressed using the CIDR notation along with other information about it.

network	geoname_id	registered_country_geoname_id	represented_country_geoname_id	is_anonymous_proxy	is_satellite_provider	postal_code	latitude	longitude	accuracy_radius
1.0.0.0/24	2077456	2077456		0	0		-33.4940	143.2104	1000
1.0.1.0/24	1810821	1814991		0	0		26.0614	119.3061	50
1.0.2.0/23	1810821	1814991		0	0		26.0614	119.3061	50
1.0.4.0/22	2077456	2077456		0	0		-33.4940	143.2104	1000
1.0.8.0/21	1809858	1814991		0	0		23.1167	113.2500	50
1.0.16.0/20	1850147	1861060		0	0	100-0001	35.6427	139.7677	500
1.0.32.0/19	1809858	1814991		0	0		23.1167	113.2500	50
1.0.64.0/20	1854383	1861060		0	0	700-0824	34.6617	133.9350	20
1.0.80.0/22	1854383	1861060		0	0	700-0824	34.6617	133.9350	20

Table 4.2: The first few lines from an IPv4 GeoLite2 file.

Each subnet is stored in a linked list mapping a subnet to its longitude. Prefixes from the previously created CSV files are matched to the corresponding subnet and the weighted average longitude for all of a subnets advertised prefixes is added to the file.

4.3 Presenting the results

4.3.1 Drawing maps

Nodes and edges

The macroscopic structure of the Internet is visualized as a circular graph consisting of nodes and edges. Each node represents an autonomous systems and each edge a connection from an AS to another. All data is based on the BGP tables collectors have stored during the first three days of the month. To ensure a reasonable good accuracy for the data, multiple collectors from different geographical locations are used to collect it. The positions, sizes and colours of nodes and edges are calculated using the following rules as suggested by CAIDA [2]:

Node's radius	$1 - log(\frac{degree(AS)+1}{maximum.degree+1})$
Node's size	$\frac{degree(AS)+1}{maximum.degree+1}$
Node's angle	Weighted average of longitudes
Link color	Node's color with smallest degree

The maps are constructed using polar coordinates, i.e. the position of a node is determined by the calculated radius and angle.

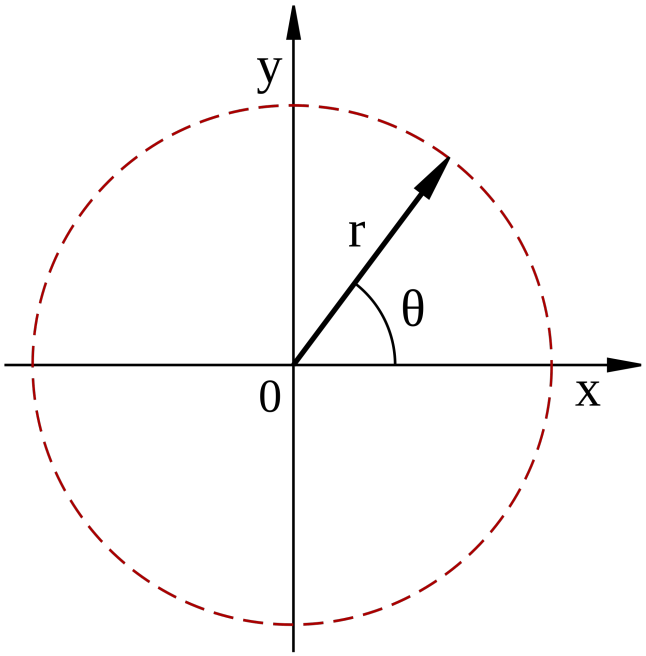


Figure 4.1: Polar coordinates
Source: `https://commons.wikimedia.org/wiki/File:Polar-coordinates.svg`

The radius $\boldsymbol{r}$ depends on the relative amount of neighbours a node has. As the difference between the nodes degree and the maximum degree of the map

$$\frac{degree(AS) + 1}{maximum.degree + 1}$$

becomes bigger, the node gets pushed further away from the centre towards the edges of the map. The AS with the most connections has a radius of 1

$$\frac{degree(AS) + 1}{maximum.degree + 1} = 1$$
$$\implies 1 - log(\frac{degree(AS) + 1}{maximum.degree + 1}) = 1$$

and is thus nearest to the centre. Similarly, the size of the node depends on the relative degree of the AS. The smaller the degree of an AS is, the closer to zero the relative size will be and the AS with most neighbours has a relative size of 1. The longitude $\boldsymbol{\theta}$ is the weighted average of all prefixes advertised by the AS.

$$\frac{\sum_i block_i.longitude \times block_i.size}{block_i.size}$$

In conclusion, each AS is represented by a square in a sector of the map corresponding to the average longitudes of the prefixes it advertises. The more connections an AS has, the larger and closer to the centre of the circle it is placed. The colour of a node depends on its degree (the amount of connections it has). Nodes with a lot of connections are large, (dark) red and located near the middle of the map, while nodes with less connections are small, blue and located near the edges.



Figure 4.2: Node colours.

The colour of an edge between two nodes depends on the size of the smaller one of them. Edges that belong to the largest nodes (in the middle of the map) are coloured in shades of red: the bigger the nodes are, the darker the edge between them is.



Figure 4.3: Edge colours for the core of the network.

The rest of the edges are slightly transparent and coloured #9bc1ff.

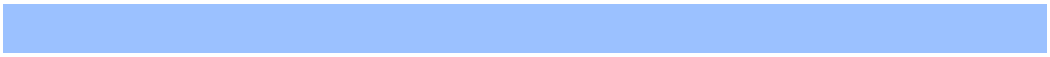


Figure 4.4: Edge colours for the rest of the network.

A complete graph with nodes and edges:

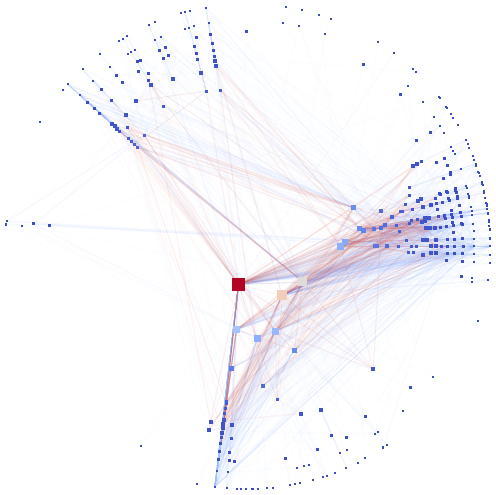


Figure 4.5: Sample graphs illustrating different colours.

Labels

The default way of labelling a map is by using labels describing the geographical location each sector of the map represents. There are six, partly overlapping sectors: Oceania, Asia, Africa, Europe, North America and South America.

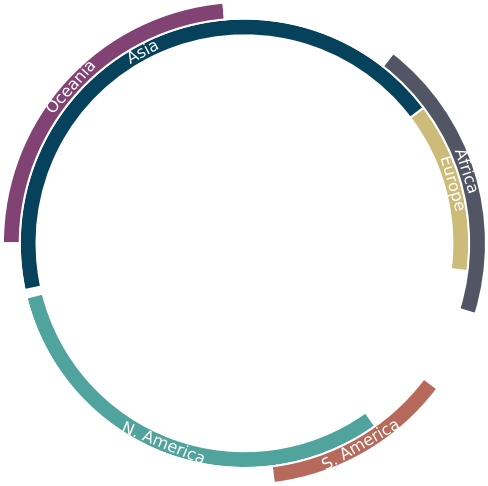


Figure 4.6: Default labels - geographical locations.

An alternative way of labelling the map is by displaying the locations of major cities on the map. Compared to the other labels, these can be used to examine the locations of interesting points on the map. Both labelling techniques are implemented, but for the sake of simplicity (and the size of the images) the first one is used for report.

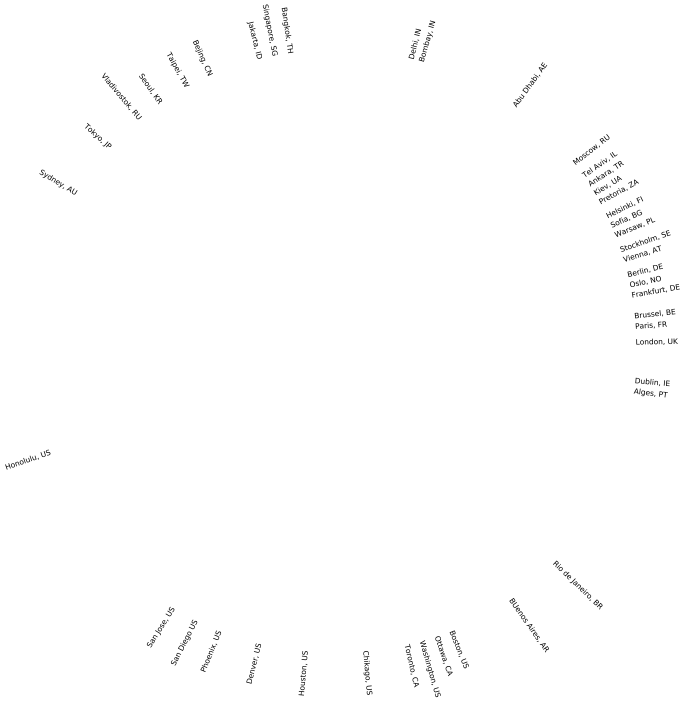


Figure 4.7: Alternative labels - cities.

4.3.2 Collecting statistics

In addition to creating visualizations of the Internet, an important part of the project is to analyse the structure behind the images. During the data collection phase, results are stored in CSV files that can be used for both drawing the maps and collecting statistics. For the purpose of the project, data is collected and compared for (the first three days of) the first month of each year, both for IPv4 and IPv6. The purpose of this is to examine the evolution of the Internet throughout the years without focusing on the minor changes from month to month. However, the software allows users to determine the exact days when data is collected, and can thus be used to analyse changes in the Internet during short periods of time.

The following information is collected:

- **Active autonomous systems**. Each autonomous system connected to a network can inform other systems of its existence by advertising the prefixes belonging to the it. BGP tables are collections of prefixes, optimal paths and other relevant information related to routing in the Internet. There are currently more than 80 000 registered autonomous systems in the Internet. However, not nearly all of them are actively advertising prefixes or routes. Furthermore, the Internet is still growing extremely fast, and the amount of active systems changes over time.

  Questions that are of interest:

  - How many autonomous systems are actively advertising prefixes and routes over BGP?
  - How fast is the amount of ASs advertising IPv4/IPv6 data growing?
  - Is there any significant difference in the activity of the two IP versions?

- **Total number of prefixes**. An autonomous system can consist of one or more subnets. The total number of prefixes being advertised gives an idea of the size of the Internet and is especially helpful in comparing the two IP versions and examining their growth.

  Questions that are of interest:

  - How much has the use IPv4/IPv6 grown in the past 15-20 years?
  - How fast is IPv6 growing?
  - Is there any significant difference in the activity of the two IP versions?

- **Average number of prefixes per AS**. The distribution of advertised prefixes per AS is not even. Some systems only advertise one (or zero) prefix, while others advertise hundreds of prefixes. The average number of prefixes per AS describes the average size of an AS. Combined with the previous graphs it helps in understanding the dimensions of the Internet. Only active ASs (≥ 1 prefix) are considered for the average due to the large amount of inactive ASs.

- **Average prefix lengths**. The way addresses are distributed amongst ISPs plays a significant role in the time it takes to allocate an entire address pool. For IPv4 this happened in 2011, and for IPv6 this is likely to never happen due to the enormous size of the address pool (2^{128}). However, if blocks of addresses are allocated too carelessly, many addresses will go unused due to the large size of single blocks. For example, the typical IPv6 block is a /48 network. This means that the end user can have 65,536 /64 subnets, i.e. a total of 1,208,925,819,614,629,174,706,176 addresses. [8]

  Questions that are of interest:

  - How large is a typical block of addresses?
  - Is the average size coherent, or has it changed over time?

- **Average degree**. In addition to the amount of prefixes each autonomous system contains, another significant factor for the structure of the network is the amount of connections (*degree*) each AS has to others. As with the amount of prefixes, the distribution of connections is fairly uneven. The core network can have closer to a hundred connections to other ASs, while the edges only have a few. Only active ASs are considered due to the large amount of inactive ASs.

  Questions that are of interest:

  - How many connections does each AS have on average?
  - Is the average degree coherent, or has it changed over time?

Chapter 5

Results

5.1 Statistics

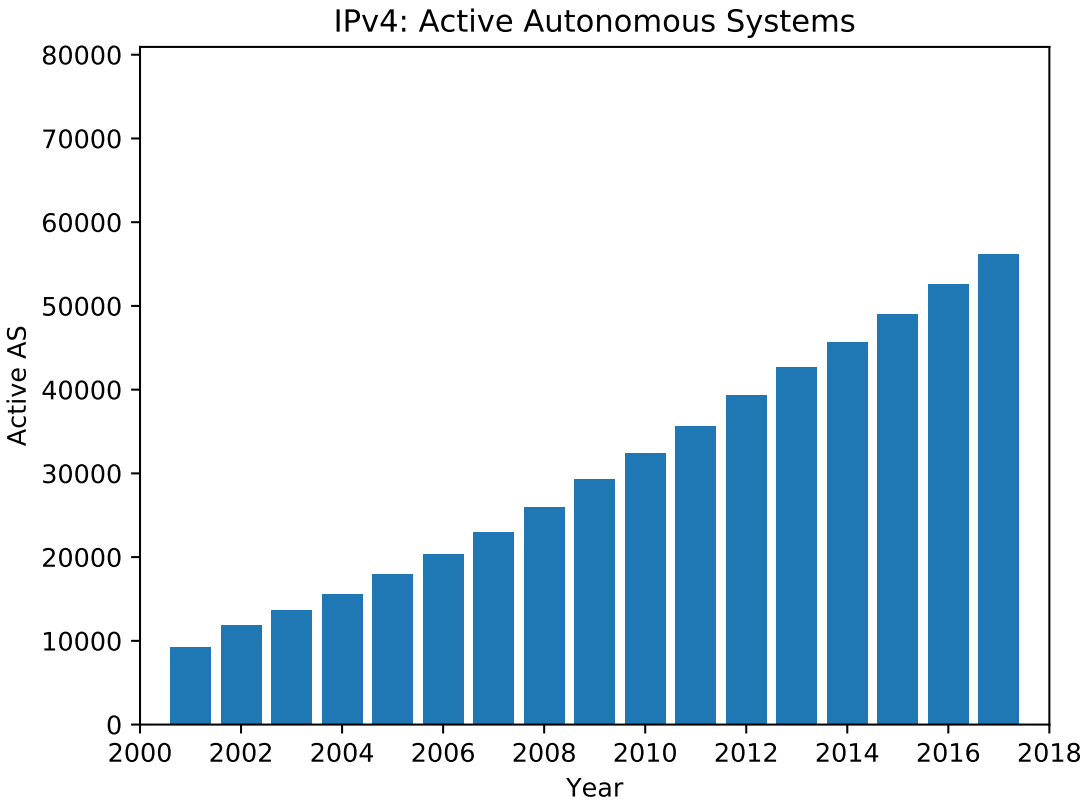
There are currently more than 80 000 registered autonomous systems. Out of these, less than two thirds (roughly 60 000) are advertising IPv4 prefixes and only a fraction (roughly 10 000) are adapting to IPv6. The amount of autonomous systems that are advertising IPv6 prefixes is currently on the level IPv4 advertising was in early 2000 (approximately mid-2002). However, the amount of IPv6 users is steadily growing, especially since 2011 when the last blocks of IPv4 addresses were allocated. The growth is likely to increase in the few upcoming years as large amounts of IP addresses are needed for new technologies, such as mobile devices, smart-TVs and the Internet of things.

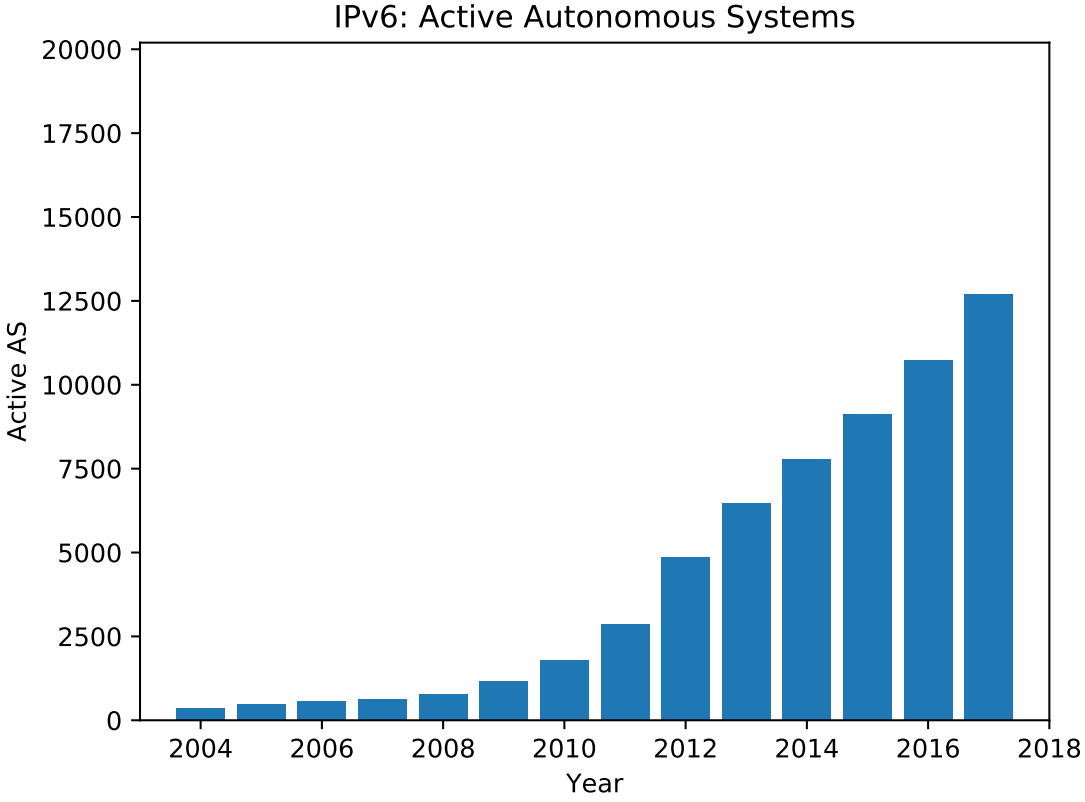
The amount of advertised prefixes is steadily growing for both IP versions. This is fairly intuitive, as the need of addresses is growing and there are still many assigned but unused addresses left to be allocated to end-users. An important observation to be made when comparing these with the previous graphs, is that the amount of advertised IPv6 prefixes in 2017 (<40 000) is significantly less than the amount of advertised IPv4 prefixes was in 2002 (>100 000) when the same amount of autonomous systems were active. In other words, although a the amount of IPv6 prefixes is steadily growing, it seems like only a small amount of ASs are using a large amount of IPv6 addresses. However, this group of early adapters consists mostly the larger ASs making up the core network, which can also be seen when looking at the IPv6 visualizations.

The average amount of prefixes per AS is relatively consistent. There is a slight increase in the averages for both IP versions, but the balance between old ASs advertising more prefixes and new ASs starting to advertise small amounts of prefixes is fairly even. However, the average amount of IPv6 prefixes has more than doubled since 2010, once again showing that a small group of early adapters is becoming increasingly active in advertising IPv6 addresses. The average prefix lengths are coherent for both IP versions. The average size of an IPv4 prefix is around 22, which equals to 1024 addresses. For IPv6, the average is roughly 38. The minimum allocation size is /32, and currently a most new blocks being allocated are of size /48 [8].

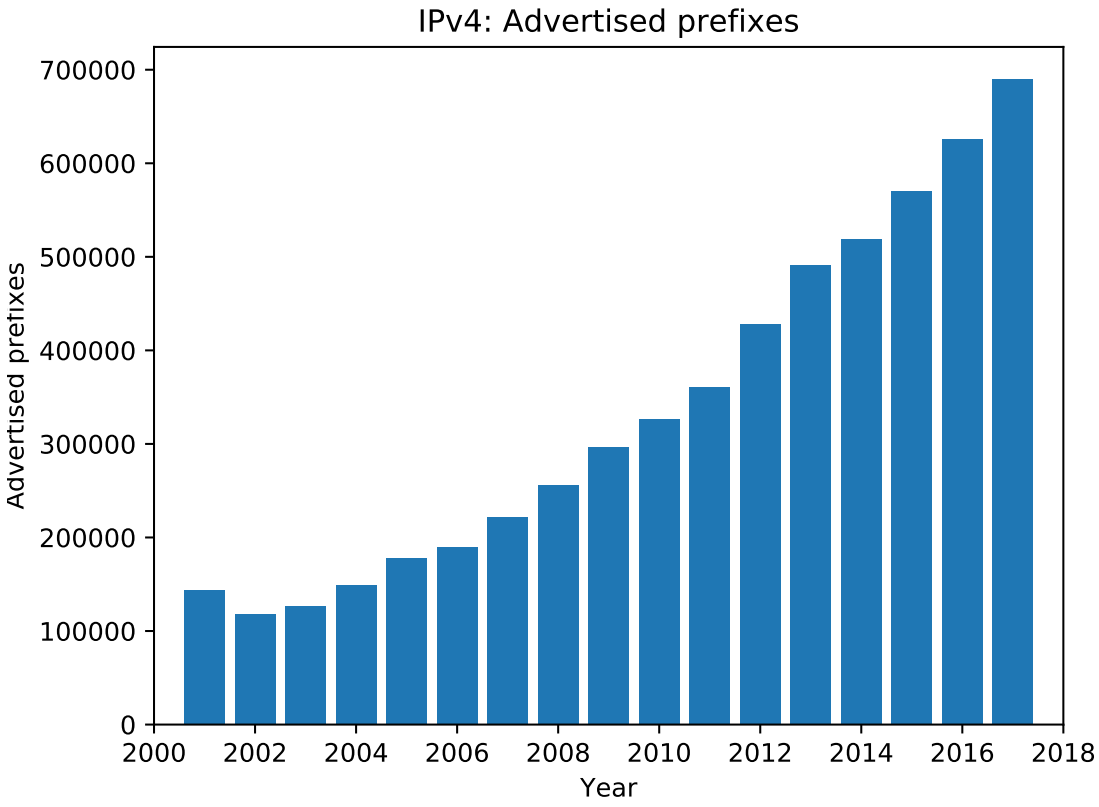
The average degree of an AS resembles the graph displaying the average amount of prefixes. New ASs are constantly joining the network, but at the same time already connected systems are getting more neighbours, keeping the average degree constant.

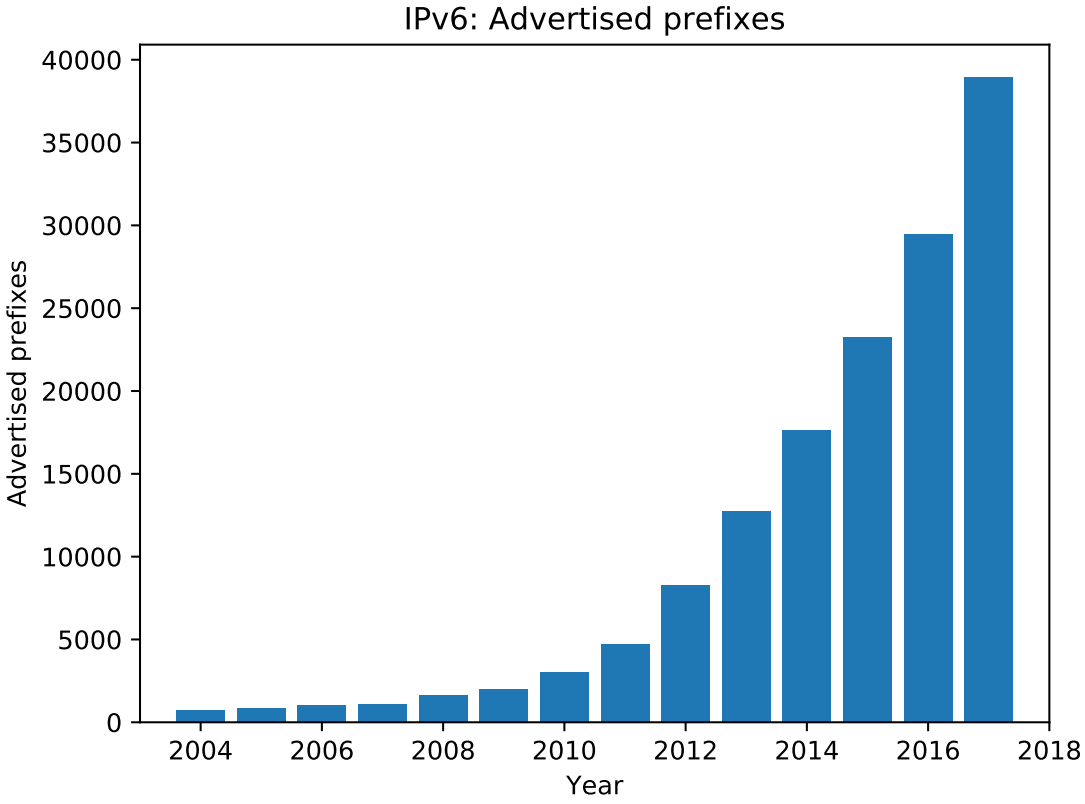
Active autonomous systems



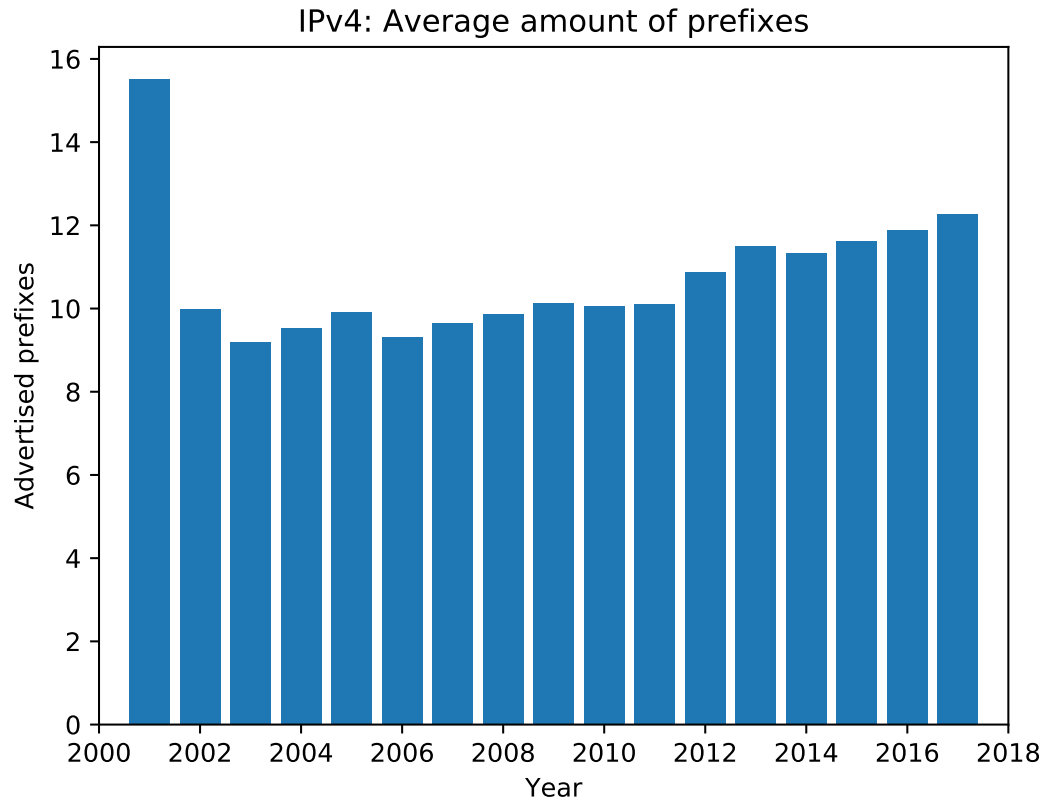


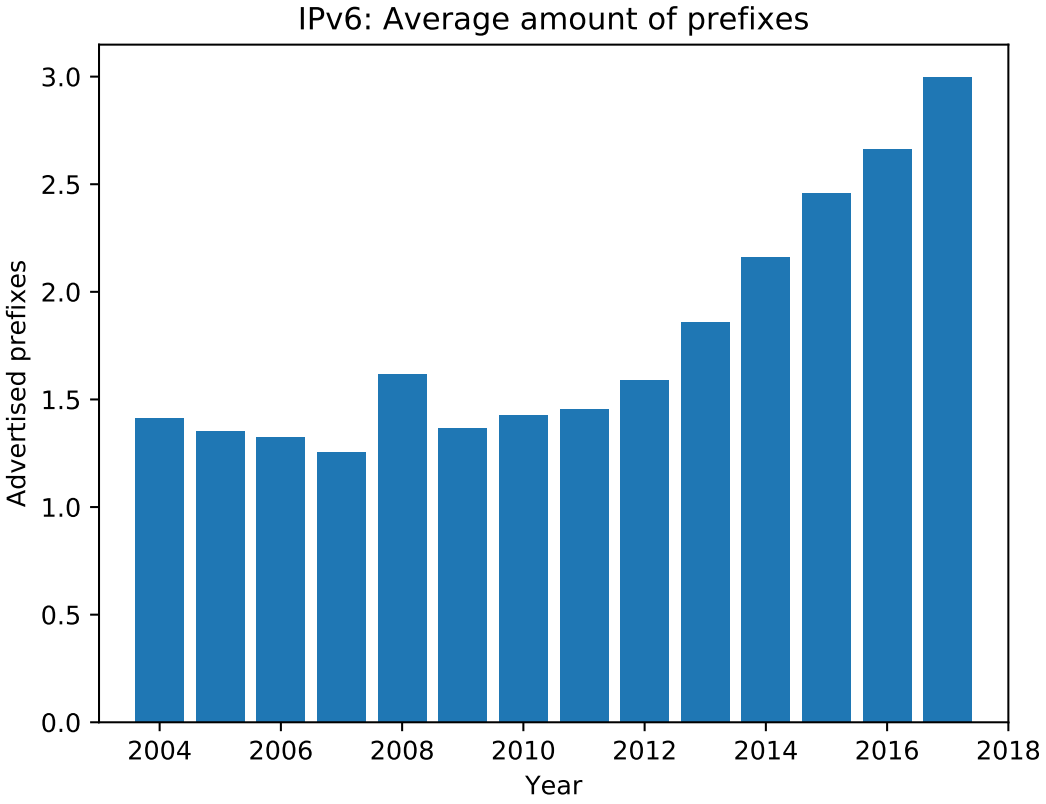
Total number of prefixes



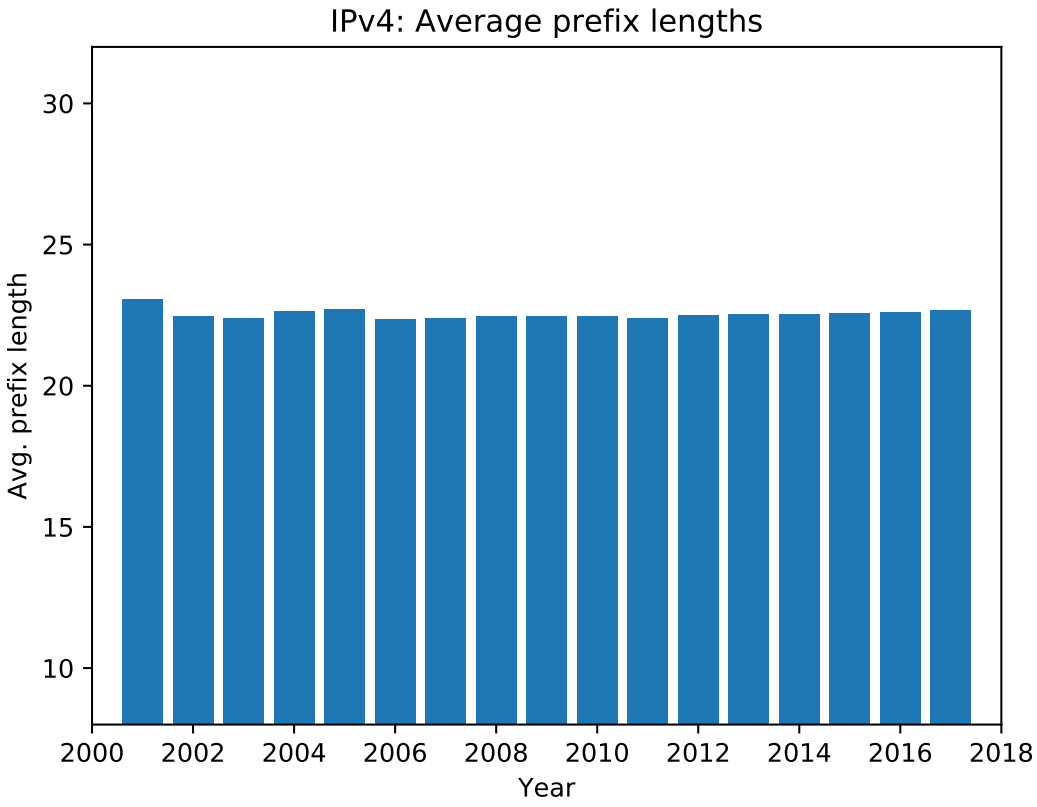


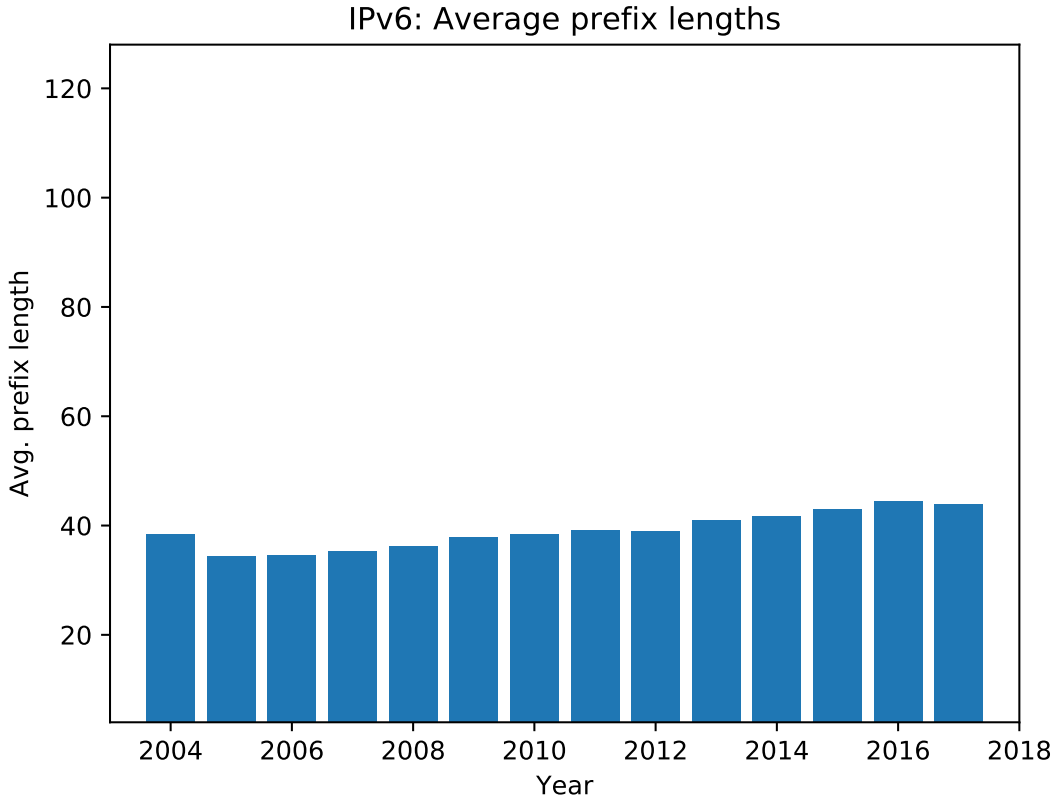
Average number of prefixes per AS



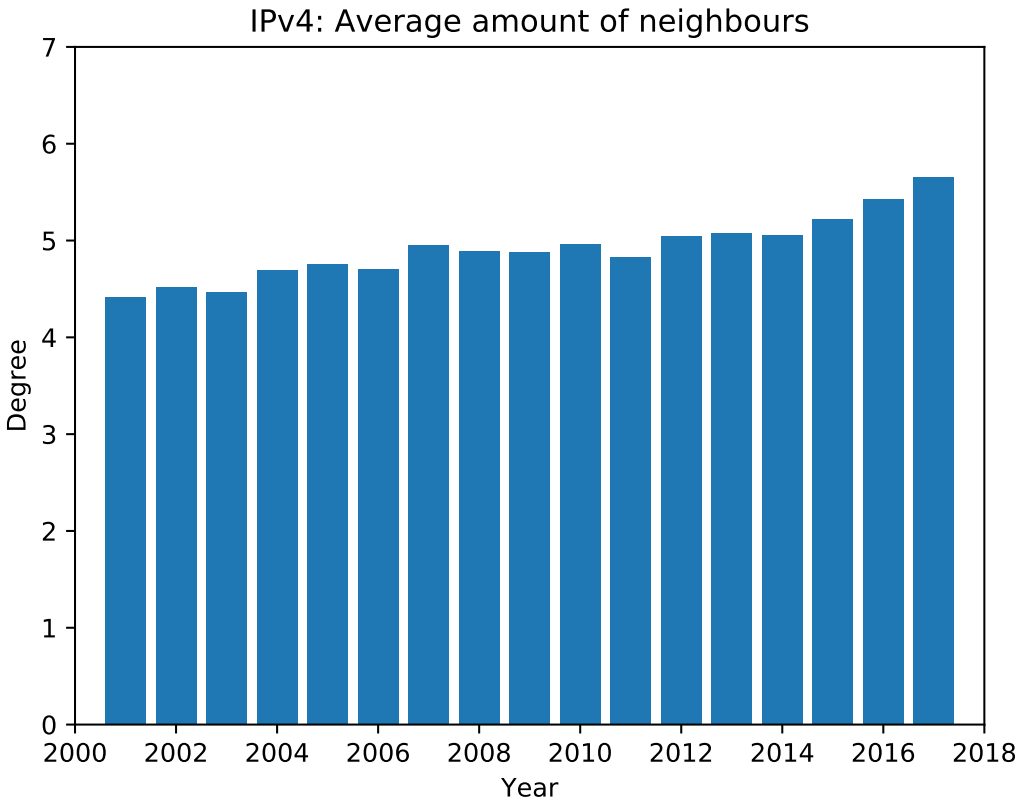


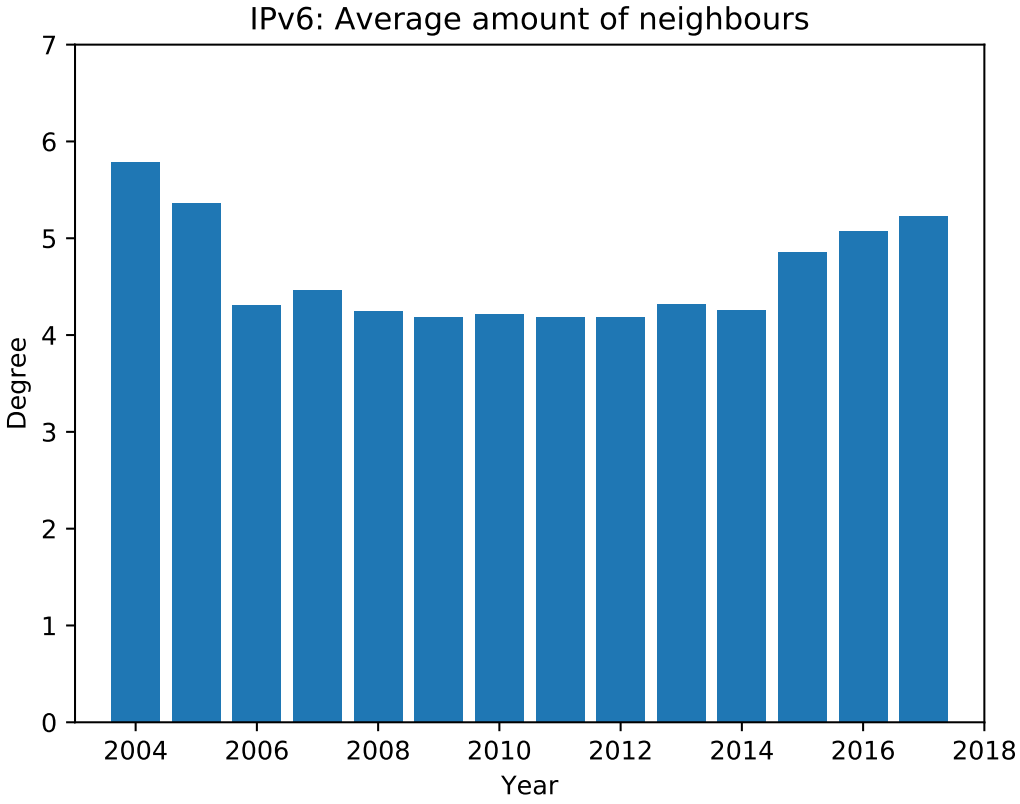
Average prefix lengths





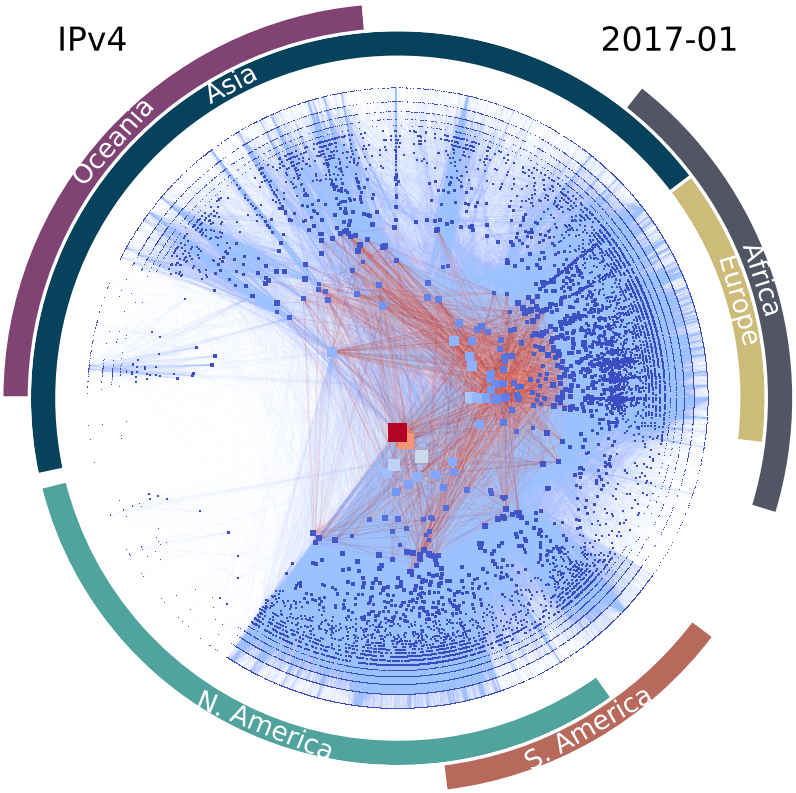
Average degree

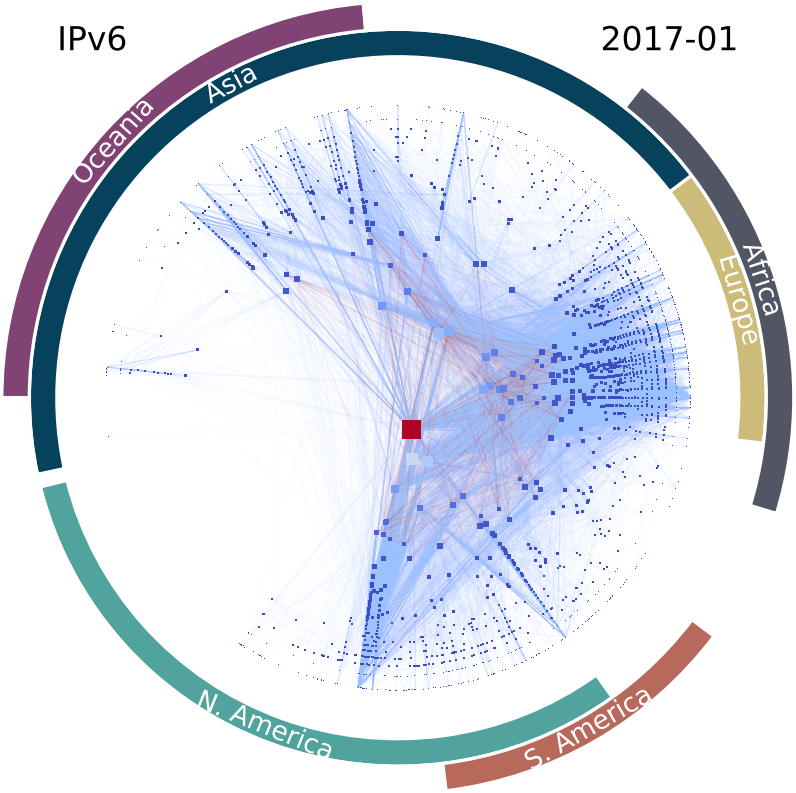




5.2 Maps

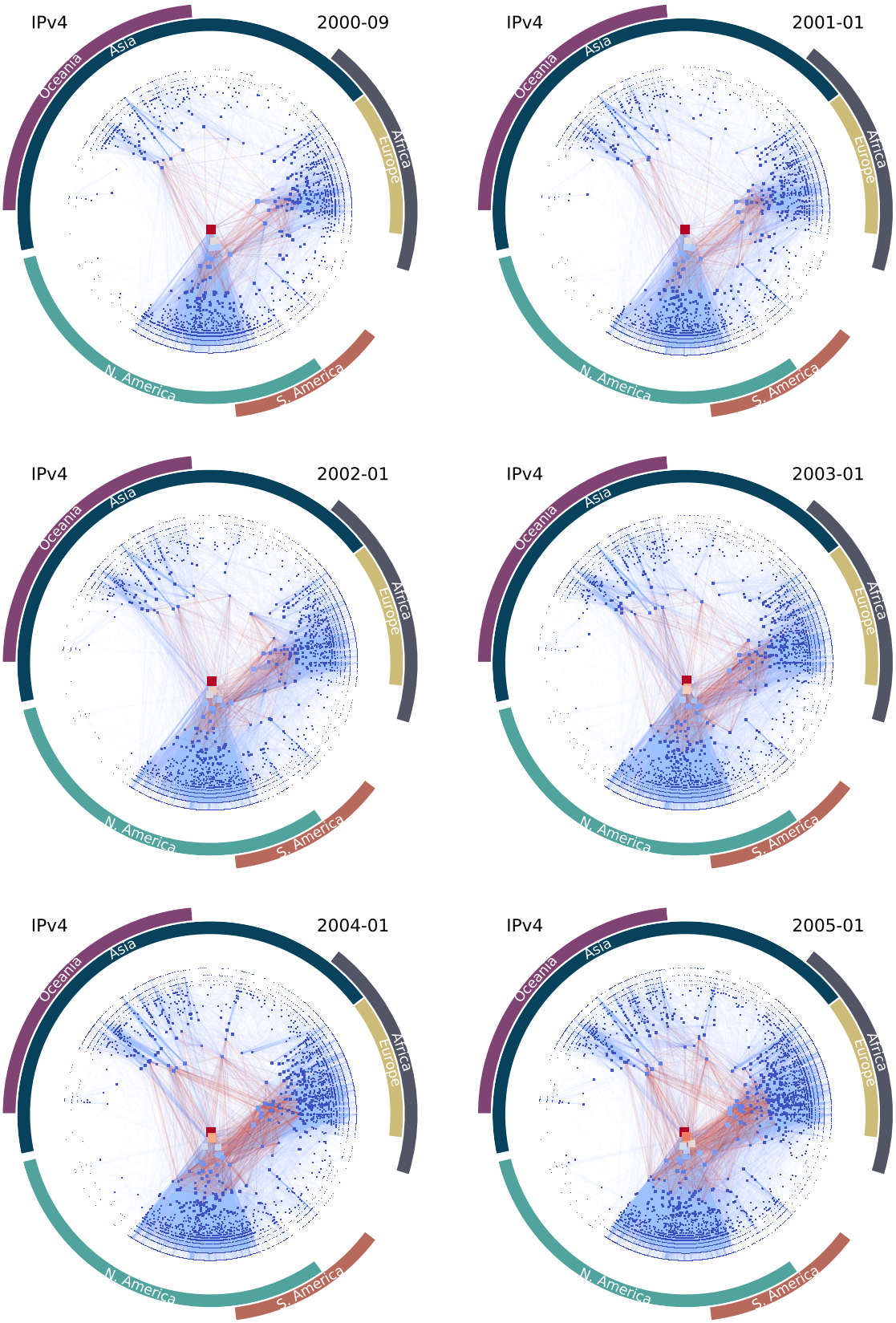
5.2.1 Recent data

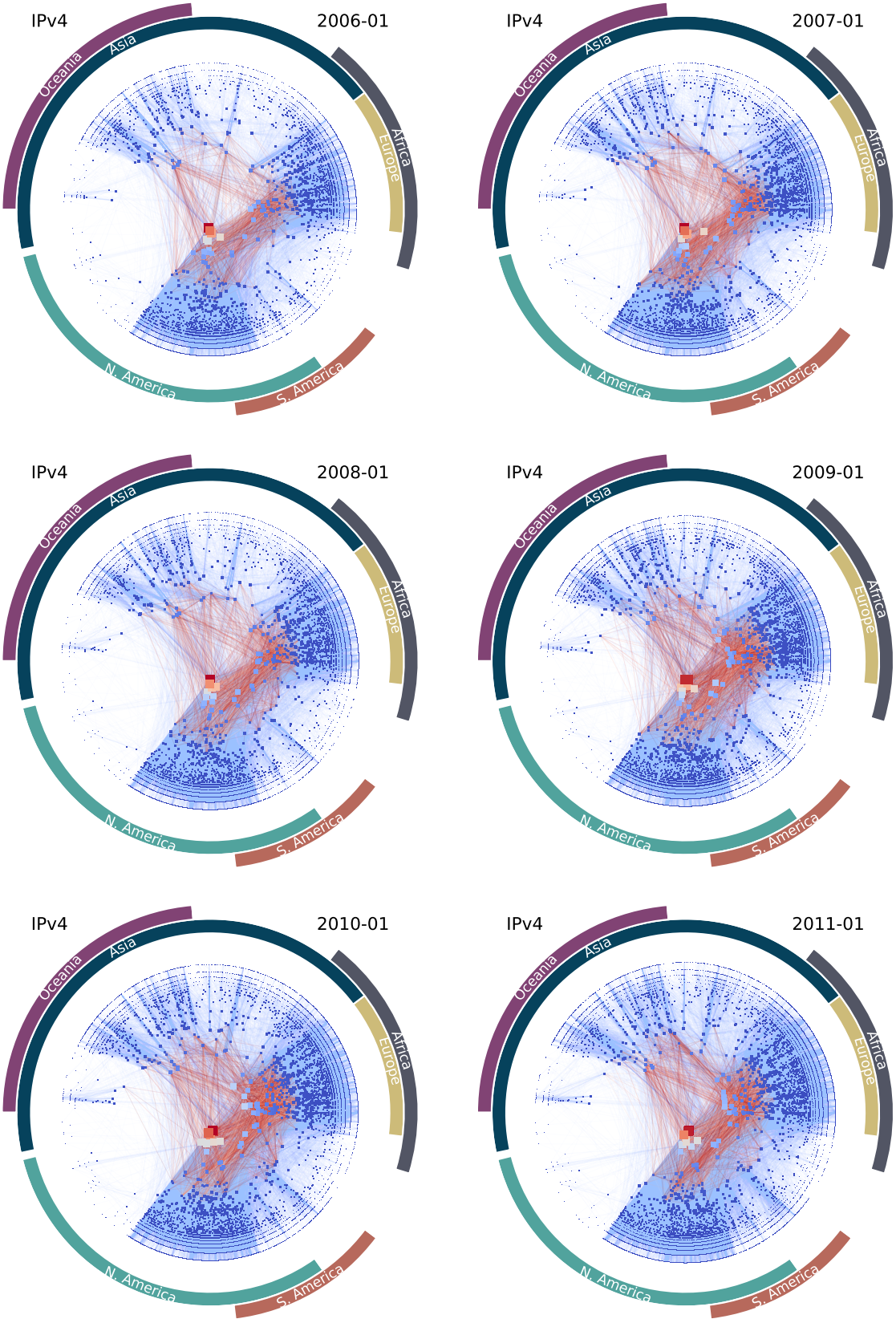


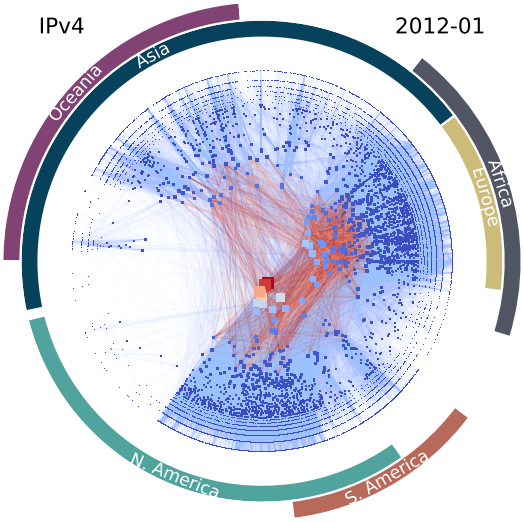


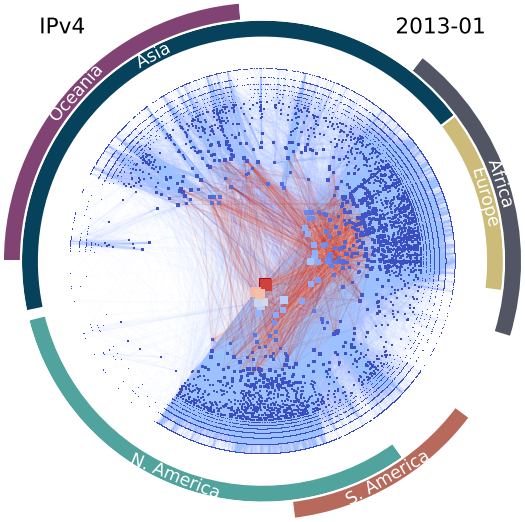
5.2.2 Historical data

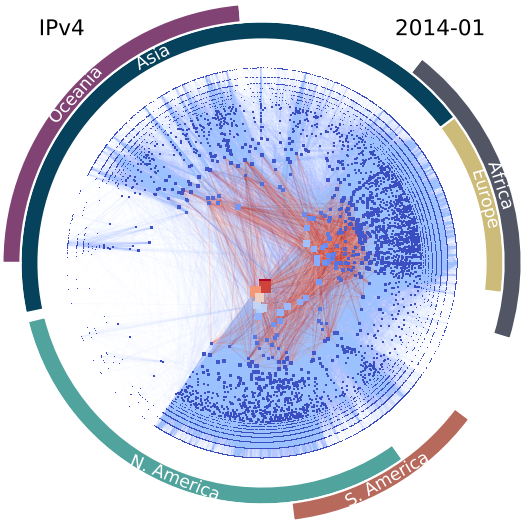
IPv4 2000-2017

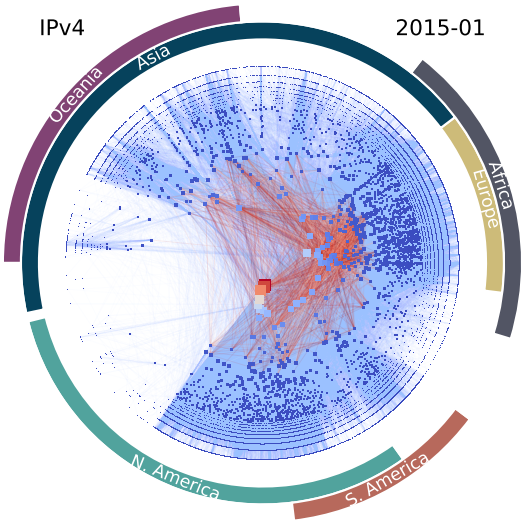


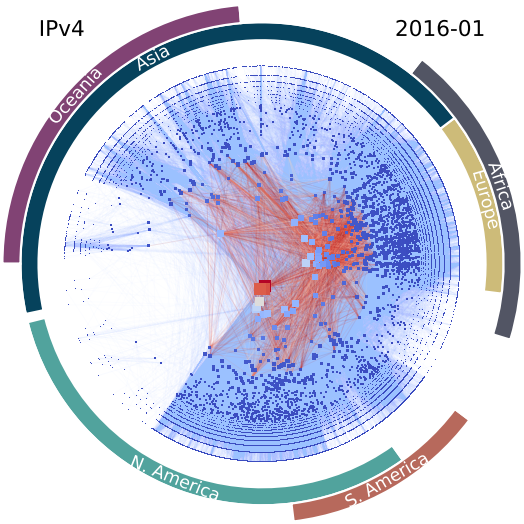

IPv4
2006-01
Oceania
Asia
Africa
Europe
N. America
S. America
IPv4
2007-01
Oceania
Asia
Africa
Europe
N. America
S. America
IPv4
2008-01
Oceania
Asia
Africa
Europe
N. America
S. America
IPv4
2009-01
Oceania
Asia
Africa
Europe
N. America
S. America
IPv4
2010-01
Oceania
Asia
Africa
Europe
N. America
S. America
IPv4
2011-01
Oceania
Asia
Africa
Europe
N. America
S. America

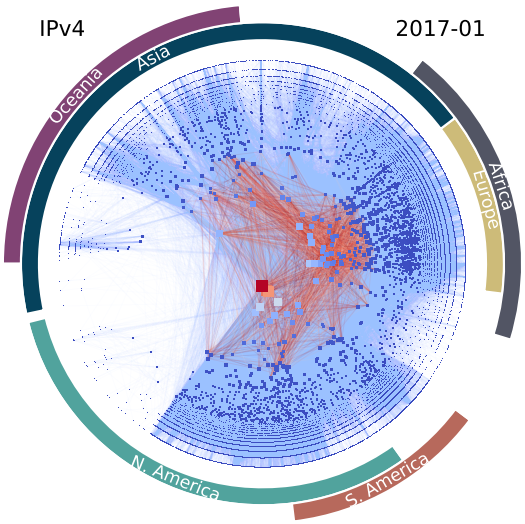

IPv4
2012-01
Oceania
Asia
Africa
Europe
N. America
S. America


IPv4
2013-01
Oceania
Asia
Africa
Europe
N. America
S. America

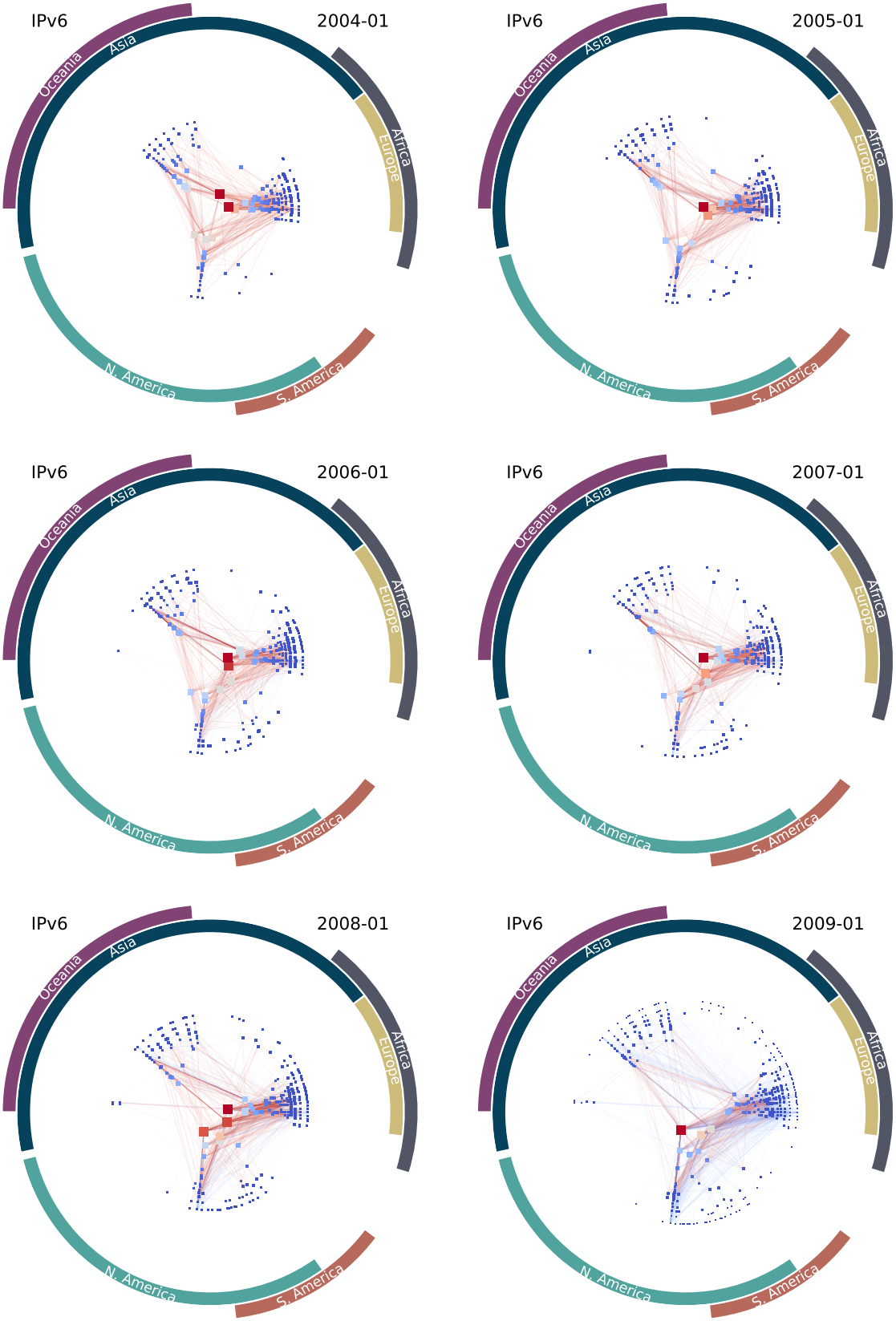

IPv4
2014-01
Oceania
Asia
Africa
Europe
N. America
S. America

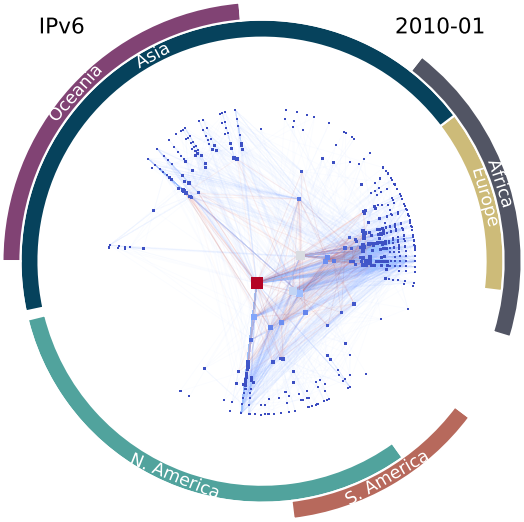

IPv4
2015-01
Oceania
Asia
Africa
Europe
N. America
S. America

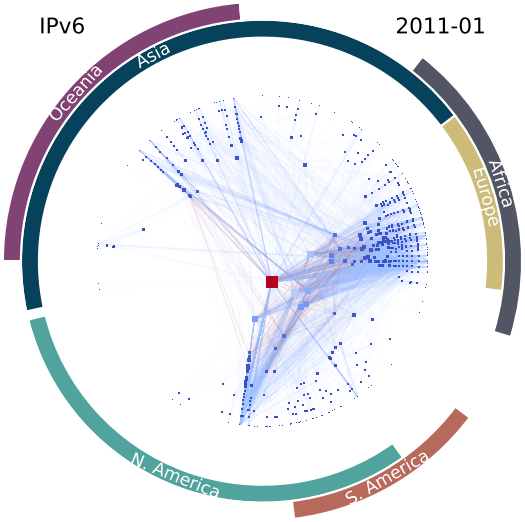

IPv4
2016-01
Oceania
Asia
Africa
Europe
N. America
S. America

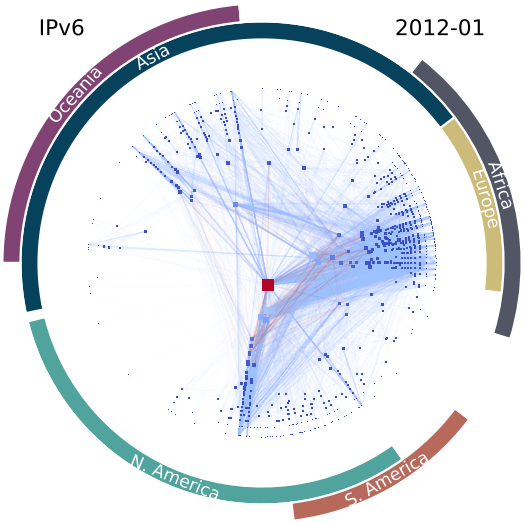

IPv4
2017-01
Oceania
Asia
Africa
Europe
N. America
S. America

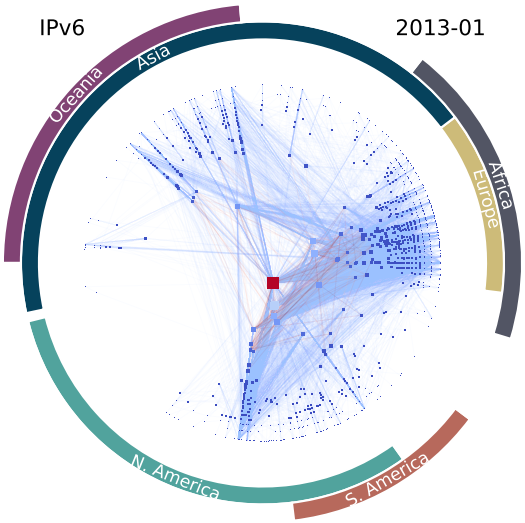
IPv6 2004-2016

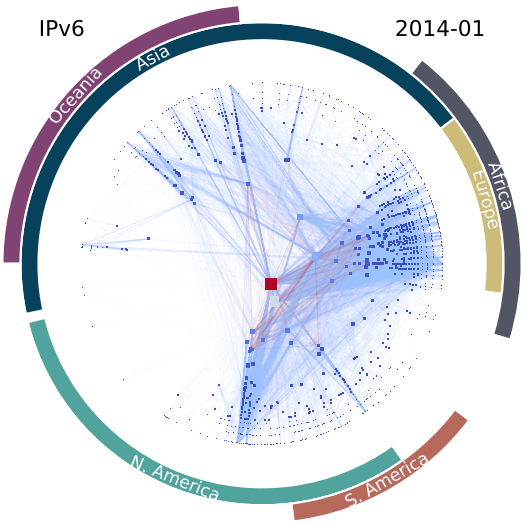


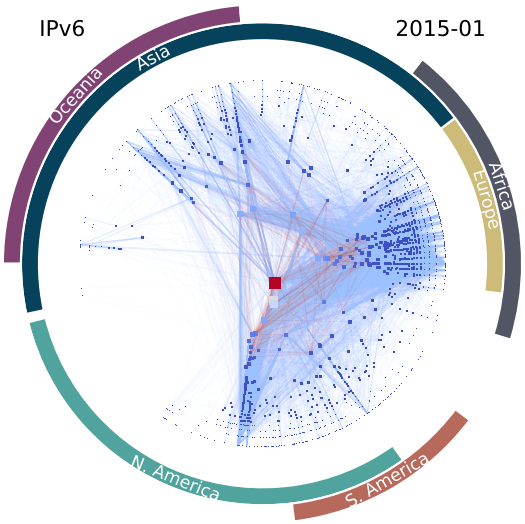

IPv6
2010-01
Oceania
Asia
Africa
Europe
N. America
S. America

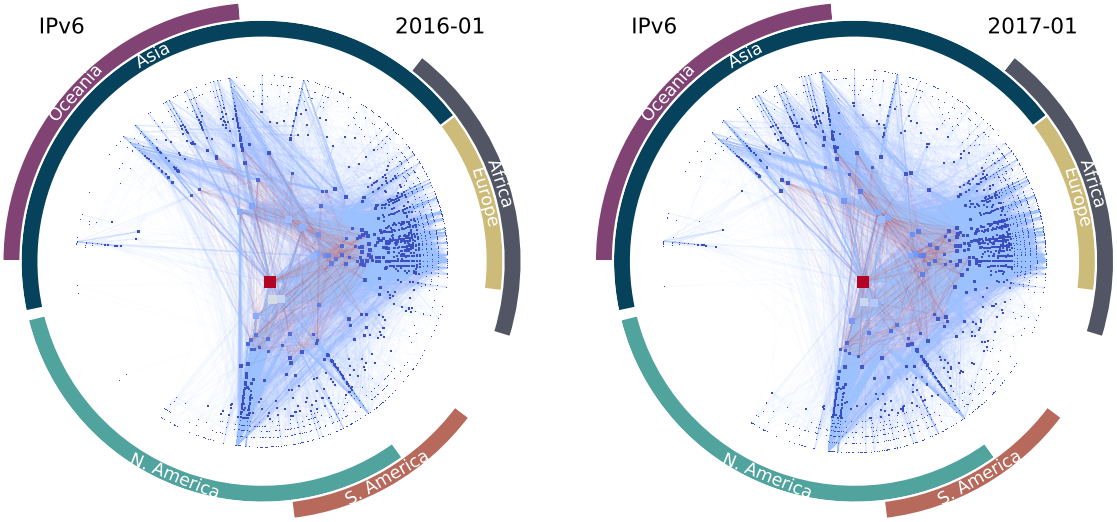

IPv6
2011-01
Oceania
Asia
Africa
Europe
N. America
S. America


IPv6
2012-01
Oceania
Asia
Africa
Europe
N. America
S. America

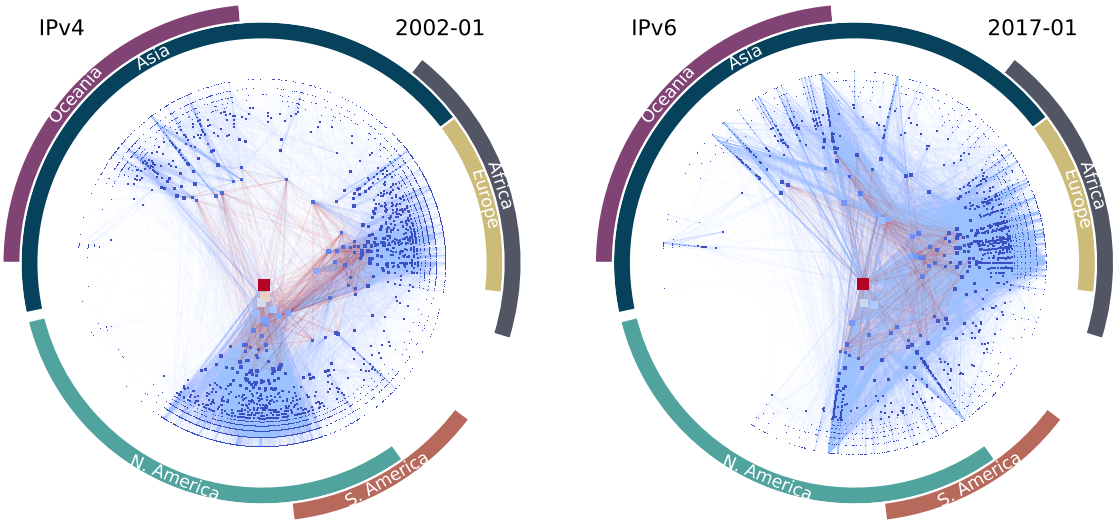

IPv6
2013-01
Oceania
Asia
Africa
Europe
N. America
S. America


IPv6
2014-01
Oceania
Asia
Africa
Europe
N. America
S. America


IPv6
2015-01
Oceania
Asia
Africa
Europe
N. America
S. America



As the statistics already showed, IPv6 is still in its early stages. The use of IPv6 prefixes seemed to be mostly experimental at first, until a wider range of ASs starts adapted the new IP version around the year 2008. In the visualizations (and especially in the animation) this can be seen as small movement of nodes until the network eventually starts visibly growing and new nodes starts appearing near the borders and the centre of the map. Statistically, the current stage of IPv6 prefixes in use is on the same level as IPv4 was in early 2000s. However, comparing the visualizations of IPv4 in 2002 and IPv6 in 2017 shows that there is a significant difference in the worldwide distribution of prefixes being used.



For IPv4 most of the major ASs in January 2002 were located in North America and Europe, while IPv6 is currently actively used in Europe and Asia. Some of the largest networks are still in America, but European ASs seem to be more active in the deployment of IPv6 compared to IPv4 in 2002. IPv4 is still slightly growing and changing due to new addresses being allocated to end-users, but the change is not nearly as radical as IPv6. As the last blocks of addresses are soon in use, the change is expected to be even less significant. This will result in a mostly coherent structure with small movement of nodes as prefixes are added and removed. However, the degrees of each node might still change even after no new prefixes are being added to the network.

Chapter 6

Development, conclusions and future work

6.1 Developing software for the project

The aim of this project was to create tools for visualizing and analysing the structure of the Internet. The purpose of these tools is to present the development of the network using simple graphs to replace the need for overly complicated numbers and tables. The project consisted of two main parts: (1) developing the software; and (2) analysing the results. The development phase consisted of four parts, each adding new features to the previous version. In addition to adding new features, each phase improved already implemented parts based on the results of further testing of the software.

The initial version of the software was a simple tool for visualizing (IPv4) traceroutes[1] using Graphviz[2]. The purpose of this exercise was to familiarize myself with BGP and parsing large data files. Although being a seemingly simple task, I struggled at first with the huge amounts of data. The length of a single `bgpdump` is currently exceeding 30 million rows (each row representing one prefix) for IPv4 and 1 million rows for IPv6, and as previously discussed, achieving the required accuracy means that the software needs to be capable of combining approximately 30-40 files from every collector for each map. For IPv4 this means going through more than a billion rows of just BGP data to draw a recent visualization of the Internet (January 2017: 1 214 000 000 lines of IPv4 data and 38 000 000 lines of IPv6 data)! Knowing this, I decided to put in a lot of effort into designing sufficiently effective and modular structures and algorithms for the initial version, in order to avoid problems later on. Furthermore, as the main software is written in C, I continuously checked it for memory leaks due to my active use of pointers and memory allocation. Although most of the initial code base was rewritten or improved a some point, the basic structure is still the same as in the original design.

The next few iterations changed the software from producing DOT files from traceroutes, into its current format, i.e. a software that collects BGP data from multiple dumps and combines it into CSV files. At this point, `bgpdump` was still used for parsing files that were downloaded directly from the collectors using shell scripts. This was both time and space consuming (especially for IPv4 as the size of a recent dump is around 115M), but still acceptably efficient for the purpose of the project. When I got my first complete CSV files, the next problem was how to visualize the results. I experimented a lot with different tools similar to Graphviz, but decided at an early stage to implement my own software using available Python libraries (see section 2.3). Deciding to split the program into multiple parts meant that I could easily run them in parallel. To coordinate and simplify the execution of different parts, I wrote a shell script (see appendix A) that controls the entire process.

[1] https://linux.die.net/man/8/traceroute
[2] http://www.graphviz.org/

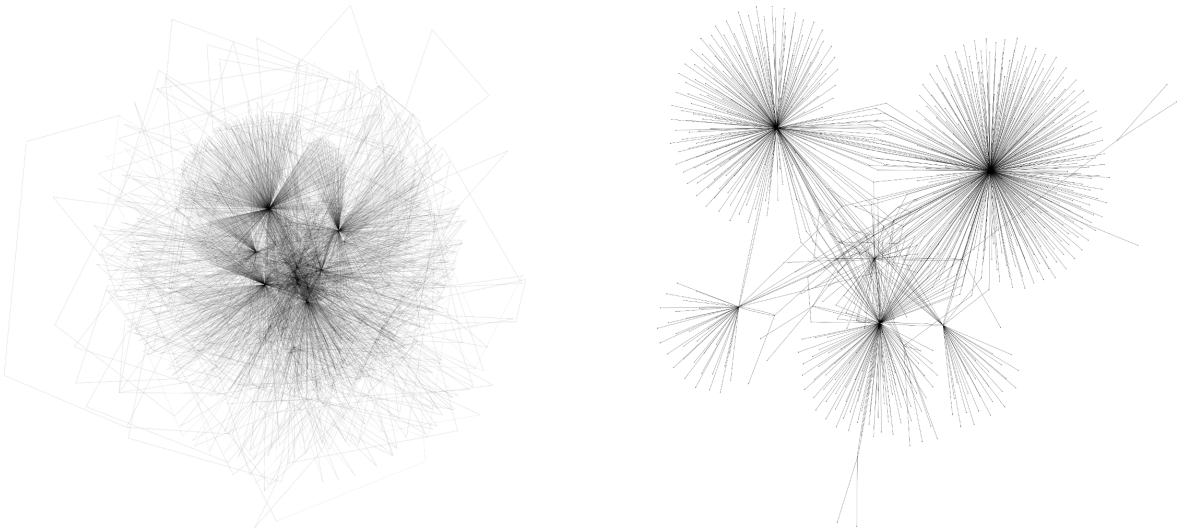


Figure 6.1: An early version of visualizations that did not support custom sizes or locations of nodes.

The next significant step was adding support for IPv6. Until this point, the software only collected IPv4 data. Thanks to the careful design of the original version, allowing IPv6 data to be collected was mostly straight forward and did not require any major changes in the software. Each part of the software was changed to take the desired IP version as a parameter and adjust its behaviour to match the format of the prefixes, the data sources and the amount of data being collected. However, finding the correct longitude based on matching the prefix to a subnet turned out to be a difficult task. The initial version of the software used a rather ineffective algorithm performing a bitwise check to find the correct subnet. An IPv6 address consists of eight groups of four hexadecimal digits, making the previous algorithm way too inefficient for comparing prefixes [6]. To solve this problem, I once again decided to divide the program into two parts and collect locations with a Python script using the `ipaddress`[3] library. The program now consisted of three parts with different tasks: (1) combine AS data with BGP dumps; (2) find the location of each prefix and calculate the approximate longitude of each AS; and (3) visualize the data.

The final major change in the software was changing from `bgpdump` to `BGPStream`. As already briefly mentioned in section 2.3, the main advantage of this change was eliminating the unnecessary step of downloading the files before parsing. Fortunately `BGPStream` was designed to replace `bgpdump`, which made the change extremely easy and did not require any major changes in the software. During the last few months of the project, I mainly collected data and made minor improvements and bug fixes as it became necessary. The software now fills the requirements presented in chapter 3 and can be used for the intended purposes.

6.2 Conclusions and future work

IPv6 has been around since late 1990s. However, up until a few years before the exhaustion of IPv4 addresses in 2011, the use of IPv6 was more experimental than practical. Even today, nearly 20 years after the standardisation of the protocol and more than 5 years after the last IPv4 blocks were allocated, only every sixth autonomous systems using IPv4 has started advertising IPv6 addresses. Predicting the growth of IPv6 is still mostly speculative as the changes have been rather slow until a few years ago. The use of IPv6 will most likely keep on growing as more networks are adapting the new protocol. Furthermore, the need for IP addresses is likely to change radically when new technologies, such as the Internet of things, are being developed. However, since the IPv6 address

[3] https://docs.python.org/3/library/ipaddress.html

pool is enormous, it is not likely to be exhausted during the lifetime of the Internet, unless the waste of unused addresses in allocated blocks gets too high.

The software developed for this project meets its requirements and is capable of collecting large amounts of data reasonable efficiently. It combines data from multiple collectors, visualizes the results, calculates statistics and presents them as graphs. However, the amount of collected data is fairly low due to limited time. Each map uses 1-4 independent data collectors of BGP tables, one collector of AS numbers and one collector of geolocations. The achieved precision is sufficient, but could be improved if more collectors were used and their results compared. Faulty data is currently discarded without further analysis and every map displays the Internet as it looked during the first three days of the month. The animations displaying the evolution of the network use one map for each month from the past 13-17 years. Older data was unfortunately not available, and creating more than one map a month would have required more time for data collecting. The coverage and accuracy of the animations could be improved by using more frequent plots to show changes as they occur on a weekly basis. The statistics currently compare the changes on a yearly level without going into further details about individual years.

In conclusion, the software works reasonably well, and the collected data gives a decent overview of how the Internet has evolved throughout the years. As discussed in the previous section, the software can be used to analyse arbitrary time intervals and is easily modified to meet future needs. Besides collecting more recent data and statistics, another interesting task would be to examine changes in the Internet during various events, such as DDoS attacks or cut-offs (e.g. Egypt in 2011).

Appendices

Appendix A

Running the Program

The program is designed to run on Ubuntu 16.04.2 LTS. However, the script `run.sh` contains a modified version for running the software on FreeBSD 10.3.

Running the program from `src/`:

```
> ./run [START YEAR] [END YEAR] [IP version]
```

This will create the following files:

```
../csv/[IP version]/year-month.csv
../plot/[IP version]/year-month.pdf
```

When uncommented from `run.sh`, the following commands will iterate through all collected data to collect statistics and draw an animation of all available plots:

```
# Collect statistics for all CSV files (IPv4 or IPv6)
# Data is stored in ../statistics/[IP version]/[file name]
python statistics.py $IPV

# Combine all drawn maps (IPv4 or IPv6) to an animation
# The animation is stored in ../plots/[IP version_start date-end date]
convert -density 400 -delay 20 -loop 0 ../plots/$IPV/*.pdf
    ../plots/animations/$ANIMATION.gif
```

Bibliography

[1] About CAIDA. http://www.caida.org/home/about/, . [Online; Accessed: 2017-03-01].

[2] IPv4 and IPv6 AS Core: Visualizing IPv4 and IPv6 Internet Topology at a Macroscopic Scale in 2015. http://www.caida.org/research/topology/as_core_network/2015/, . [Online; Accessed: 2017-02-15].

[3] AS Names. http://www.cidr-report.org/as2.0/autnums.html. [Online; Accessed: 2017-03-01].

[4] Autonomous System (AS) Numbers. https://www.iana.org/assignments/as-numbers/as-numbers.xhtml, . [Online; Accessed: 2017-03-01].

[5] IANA IPv4 Address Space Registry. https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml, . [Online; Accessed: 2017-03-01].

[6] Number Resources. https://www.iana.org/numbers, . [Online; Accessed: 2017-03-01].

[7] GeoLite2 Free Downloadable Databases. http://dev.maxmind.com/geoip/geoip2/geolite2/. [Online; Accessed: 2017-03-01].

[8] Understanding IP Addressing and CIDR Charts. https://www.ripe.net/about-us/press-centre/understanding-ip-addressing, . [Online; Accessed: 2017-03-01].

[9] IP Address Exhaustion And Internet Growth. https://www.ripe.net/publications/ipv6-info-centre/about-ipv6/ipv4-exhaustion/history-of-ipv4-and-ipv6-infographic/, . [Online; Accessed: 2017-03-01].

[10] Keith W. Ross James F. Kurose. *Computer Networking: A Top-Down Approach, 6th Edition*. Pearson, 2013.

[11] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. BGPStream: a software framework for live and historical BGP data analysis. In *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 429–444. ACM, 2016.