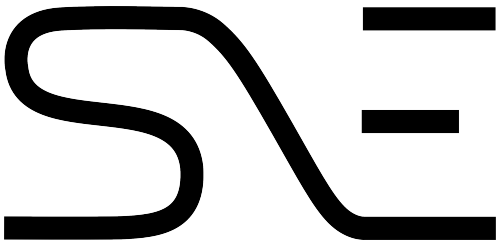

MSc Security and Network Engineering

(SNE/OS3)

Research Project 1

Network Peering Dashboard for SURFnet

by

David Garay

February 10, 2019

6 ECTS

January 7 - February 10, 2019

Supervisors (SURFnet):
Marijke Kaat
Jac Kloots

Supervisors (UvA):
Prof. dr. ir. Cees de Laat

University of Amsterdam

Graduate School of Informatics

***Abstract*—More networks are being connected every day to the Internet, network topologies are constantly changing and existing peerings need to be revisited to accommodate the ever changing user behaviour. Managing BGP peerings is a critical part of ISP activities, and its optimisation is a continuous process that requires insight into the Internet topology and constant monitoring. In this paper we work with SURFnet (the National Research and Education network in The Netherlands), and set out to answer the questions:*Which methods are available for the representation and processing of the peering relations and make optimisation recommendations? What information and which information sources should be available as input for a tool to fulfill SURFnet's requirements? Can these methods and tools also recommend peers for the best redundancy?* With the help of concrete optimisation scenarios formulated by SURFnet, we characterise the problem and propose a constraint-based recommendation system to provide BGP optimisation recommendations. We outline the design and build a recommender prototype applied to the BGP peering optimisation domain, and we identify and collect the information required. Our results shows that the application of constraint-based recommendation systems to the BGP peering optimisation domain is viable, even while acknowledging the limitations of our public domain data.**

***Index Terms*—BGP peering optimisation, Constraint-based Recommendation Systems, SURFnet**

1 Introduction

SURFnet is the National Research and Education network and among other services it provides internet connectivity to research and higher education in the Netherlands. To keep operating and improving its services, it requires a good overview of external connectivity and all of its peers.

As an Internet Service Provider (ISP), SURFnet peers with other ISPs to provide connectivity services to its customers. The roles between the peering ISPs is often defined [1][2] as provider, customer or peer; to indicate whether an ISP receives payment for the transit service, pays for it, or when a quid-pro-quo settlement takes place.

Border Gateway Protocol (BGP) [3] is used for the technical realisation of the peering relations. The cost, redundancy and performance of ISPs services are influenced by policies established with BGP attributes. There are other attributes, however, that also affect these factors: for instance, the roles of peering entities, aggregated traffic volumes (and their respective cost), and geographical information (e.g. presence at Internet Exchange Points - IXPs and Points-of-Presence - PoPs). Ignoring these parameters can lead to sub-optimal routing and lack of redundancy. For example, existing peering between organizations might need to be expanded with new BGP sessions, to avoid single points of failures. When the effort of maintenance is no longer justified by the volume of traffic, peering relationships might be stopped. Furthermore, an ISP might want to move traffic from provider ISPs to an existing or a new peer ISP, in order to improve redundancy or bring transit costs down.

The aim of our research is to help implement and build a peering dashboard to support SURFnet to visualize, recommend and report peering optimisation opportunities, based on internal or external data sources (such as http://peeringdb.com), and which is integrated with SURFnet's ticketing tool and automation environment.

Our paper is structured as follows:

In section 2 we formulate our research questions, and we briefly review related research in section 3. Section 4 provides background information.

In section 5, we evaluate our problem and propose an approach to solve it. To validate it, we design and build the prototype, and define the information required and their data sources in section 6.

We present our results, discuss them and conclude in sections 7, 8 and 9.

2 Research question

To perform BGP peering optimisation recommendations, we need to identify the techniques to process the information as well as the information required. Our research questions are structured to address these points. Additionally, SURFnet is interested in optimising a variety of peering aspects, including redundancy.

- *Which methods are available for the representation and processing of the peering relations and make optimisation recommendations?*
- *What information and which information sources should be available as input for a tool to fulfill SURFnet's requirements?*
- *Can these methods and tools also recommend peers for the best redundancy?*

3 Related work

We consulted The Center for Applied Internet Data Analysis (CAIDA) [4] to better understand the recent developments, related academic work and relevant sources of information on Internet topology. Data analysis based on, among other sources, PeeringDB [5], are studied in [6]. In particular, the graph representation of the Internet topology is discussed in [7] and [8]. They helped us understand previous work and the status of the available and relevant data sources, tooling, techniques and information.

We looked at Recommender Systems (RS) as a type of filter system able to address our problem. The work in [9] provided us with guidance on selecting the approach and [10] gives an example of a constraint-based RS.

For the prototype, data modelling and performance improvement alternatives, we looked into [11] and the PNDA framework [12].

Finally, we consulted [13][14] internal information sources relevant to the Network Peering Dashboard within SURFnet.

4 Background

SURFnet (AS1103) routers used for peering with other organizations are present at Equinix Amsterdam and Interxion Science Park. They have connections to five Internet eXchange Points (IXPs), namely: AMS-IX[15], BNIX[16], LINX[17], NL-IX[18] and Asteroid Amsterdam[19]; where it maintains peering relationships with other networks.

Peering between ISPs is becoming increasingly relevant for ISPs to, among other things, counterbalance the effects of traffic asymmetry caused by e.g. Content Delivery Networks and increasing video traffic[20]. The routing policies of SURFnet are discussed in more detail in the following section.

4.1 SURFnet Routing Policies

ISPs define policies to achieve certain goals, for instance: avoid paid transit networks, or route customer traffic directly to the customer routes. These policies are enforced by means of attributes of the BGP protocol, which influence how routing information is installed into the routers and the traffic routed.

The routing policy at SURFnet [14] is listed below:

1. Customer
2. Private Network Interconnect (PNI) - Research / Commercial
3. Bilateral: (AMS-IX, NL-IX, LINX, Asteroid)
4. Route Servers
5. Upstream/Transit

What this policy establishes is an order of preference for the routing of traffic.
The definition of a policy demands careful design and constant monitoring, to ensure that they are enforced. To support the selection of the optimal set of peers and network topologies, according to SURFnet's policies, we will see in Chapter 6.2.1 that we require the collection, visualisation and monitoring of AS attributes that are not available via BGP.

4.2 Reference Architecture

In Figure 1 we depict the relations between SURFnet and other organizations, both at organizational and BGP session level. A direct exchange of prefixes between organizations might also be realised via *Route Servers* (RtS). A route server is part of the IXP infrastructure, and its function is to advertise networks from other peers at the IXP. This permits ASes to learn network prefixes from other ASes via only one peering (that of the RtS).

The Operation Support Systems (OSS) manage, among other things, the collection of network statistics.

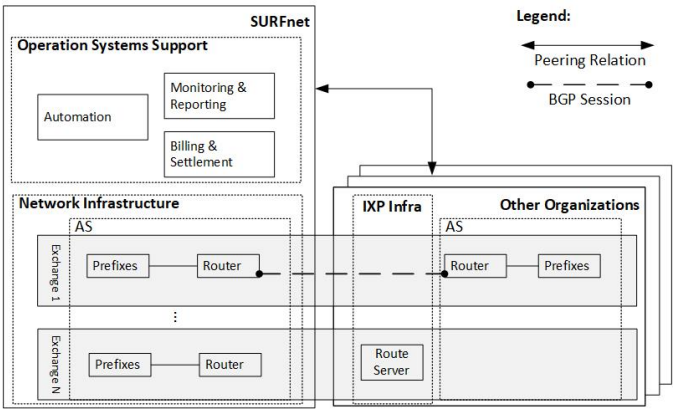

Figure 1. Reference Architecture

4.3 Current BGP Peering Process

Traditionally, ISPs identify potential peers based on, for instance, forecasted traffic volumes. If considered interesting, the ISP will then evaluate economic, geographic and capacity related constraints. If these constraints are satisfied, the peering is proposed. The operators will then monitor various peering indicators (e.g. performance, traffic volume and traffic ratio) to assert whether the peering is beneficial.

4.4 BGP Peering Process Optimisation

In [20] the authors investigated the complexities of BGP peering and made an attempt to model optimal BGP peering selection formally as a combinatorial optimisation problem and solving it. The attempt was deemed infeasible. The authors conclude that this is due to the limited ability to predict traffic, utility (due to the complex cost structure) and computational infeasibility of identifying the optimal set of peers, because of the network structure. Peering optimisation remains, however, a continuous and important process for ISPs. SURFnet, in particular, is interested in the peering optimisation scenarios in Table 1.

	Description
1	Propose suitable new peers, sharing at least one IXP with SURFnet
2	Propose the establishment of BGP sessions when missing on a router
3	Propose migrating traffic handled by route servers to a new peer
4	Propose disconnecting peers when traffic is no longer significant

Table 1. SURFnet scenarios for peering optimisation

In scenario 1, the objective is to classify and propose ASes available in at least one IXP where SURFnet is present. This could allow SURFnet, for instance, to transfer traffic from an upstream provider to a new peer.
Scenario 2 looks at BGP sessions: SURFnet expects BGP sessions to be configured on every router in an IXP shared with the peered network, to ensure redundancy.
In scenario 3, if enough traffic volume (i.e., above a certain threshold) is sent to non-peer prefixes learnt via a route server, then a direct peering will improve the reliability of the network (as the same information is now flowing through less components) and service (since when a problem occurs, a peering organisation will typically handle it with a higher priority).
Finally, in scenario 4, the number of peering relationships is kept manageable by removing peering relationships that do not comply with the peering conditions.

5 Methodology

In this section we outline our approach to map the BGP peering optimisation scenarios formulated by SURFnet to a potential solution.

5.1 Problem Characterisation

We refine the analysis of the scenarios in Table 1. More concretely, Figure 2 illustrates the filtering conditions for scenarios 1 and 3. Here, we are interested in the intersection between the subset of ASes not peering with SURFnet AS1103, and ASes sharing at least one IXP with AS1103. Additionally, for scenario 3, we will filter for ASes whose prefixes are learnt via route servers.
We highlight the challenge of identifying prefixes learnt via route servers for scenario 3: we cannot rely on identifying the route server's AS in the AS path as these are not prepended [21]. We will propose in Chapter 6.1.3 filtering rules that implement the constraints of this scenario.

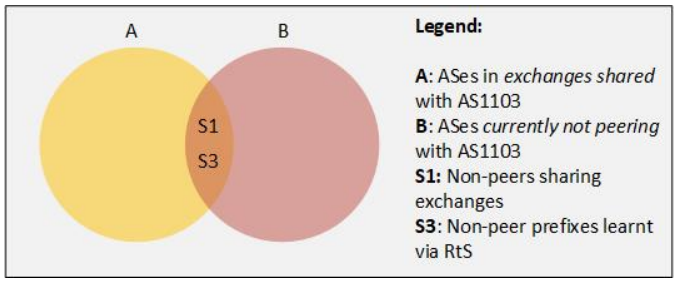

Figure 2. Analysis of scenarios 1 and 2

In Figure 3 we illustrate scenarios 2 and 4, where we filter ASes not peering with SURFnet, which share at least one IXP with AS1103.

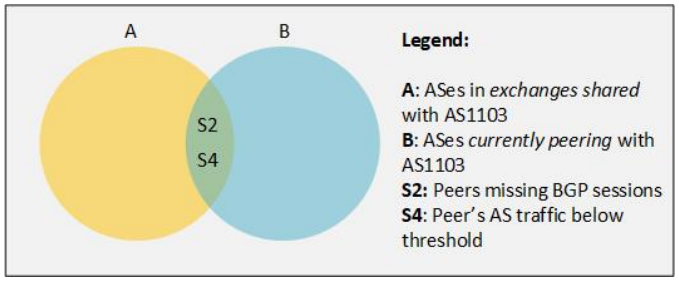

Figure 3. Analysis of scenarios 2 and 4

In general, these scenarios take the form of a set of constraints and conditions which return a set of suitable ASes when satisfied. To filter information based on constraints, we need first to identify and collect the attributes that will describe the ASes, and use the proper filtering approach to obtain the set of recommended ASes.

5.2 Recommendation Systems

RSs are a type of information filtering system, like search engines. While a search engine's main task is to locate documents that are relevant to the user's need, a recommendation system suggests items to a user that is likely to be of his interest. It does so by making a prediction of its utility[22]. Categories of RSs are Collaborative-Filtering (CF), Content-Based (CB) and Knowledge-based (KB); each one presenting their own advantages and disadvantages [22]. Figure 4 compares the different RS approaches.

1. **Collaborative-Filtering**: these systems rely on knowledge about other users and their opinions (typically expressed by a rating attribute). The definition of which group of users should be considered is based on a similarity-metric, which might rely on demographics or use other classification techniques. Problems associated with CF filters are related to cold-start (e.g. no ratings available for new products), data sparsity (not all items are evaluated) and scalability: i.e., demanding a large quantity of computing power.
2. **Content-Based**: here, knowledge is extracted from previously selected items, and used to find new similar ones. The criteria for similarity is based on the comparison of item attributes. A problem faced by this approach is to recommend new unexpected items (this characteristic is known as *serendipity*) and is a relevant criterion for RSs evaluation.
3. **Knowledge-Based**: here the RS has knowledge on how an item meets the user requirements. These requirements are typically explicitly elicited, as opposed to other approaches which extract their knowledge implicitly. **Case-based** and **Constraint-based** RSs are two well-known approaches to knowledge-based systems. In Case-based systems items will be recommended based on a domain-specific similarity criteria; while a constraint-based system will take into account explicitly defined constraints. Should no items be available that fulfill the user requirements, changes to the requirements will be proposed to the user[10].

Note that combinations of these approaches are also possible.

After item recommendations have been performed, user requirements can be further refined by means of *critiques*. The RS might suggest additional filters to support this functionality. Similarly, dealing with unfulfillable/too loose user requirements demand for the introduction of two other related functionalities: *suggesting alternative attributes* and *query tightening*.

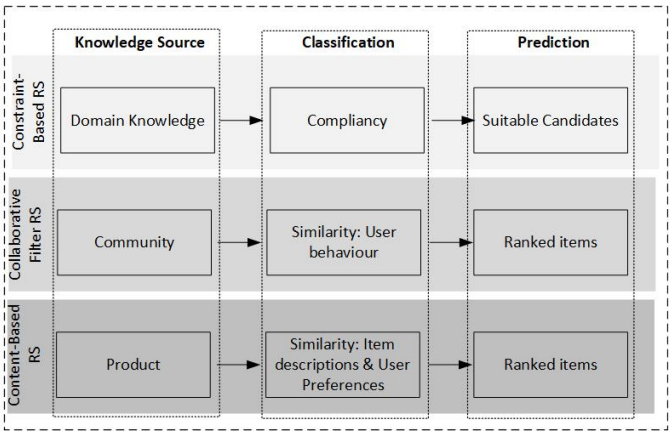

Figure 4. Overview of Recommender System approaches

5.3 Selecting a Recommendation System Approach

In [9], the author provides guidance for the selection of the most suitable RS approach based upon the domain attributes and knowledge source. We propose an RS that will recommend ASes from a dataset containing a minimal set of attributes required to address the SURFnet's scenarios in Table 1. We use these attributes to characterise our domain as follows:

- **Heterogeneity**: a heterogeneous item space contains elements with many different characteristics, while a homogeneous one will not have such variance. In our domain, the collection of ASes in the dataset is *homogeneous* (i.e. ASes and their features are similar).
- **Risk**: this aspect denotes the amount of risk a user incurs in accepting a recommendation, and determines the user tolerance. Regulatory and mandatory rules that must be obeyed are also considered here. In the domain of peer selection, making a wrong recommendation would waste the user's time and decrease the system's credibility, making it a high *risk* recommendation.
- **Churn**: churn characterises the time span of the item's relevance. If the relevance is based on opinions, new

items might not have been seen making the data set sparse. When referring to ASes and their attributes, the churn is low (they don't change frequently).
- **Preference Stability**: is used to signify whether user preferences vary with time. Stable preferences might justify collecting implicit information across sessions to build a profile. In our case, previously selected items (ASes) are a relevant source of knowledge. This characterises a domain with *stable* preferences.
- **Scrutability**: this requirement is relevant when the user requires knowing "why was this item recommended?". In our domain, we do want to have information on the fulfillment of constraint when receiving a BGP peering recommendation.
- **Interaction Style**: This aspect relates to the human interaction, and whether user requirements are implicitly or explicitly extracted. SURFnet requires the ability to define sophisticated rules that depend on specialised domain knowledge. These rules and domain knowledge are expected to be *explicitly* collected from the user.
- **Knowledge source**: main source of knowledge for the RS. Can be *social knowledge* (if predictions about individuals are extracted from peer opinions); *Individual* when knowledge comes from e.g. user ratings, but also from explicitly input requirements; and finally *content knowledge*: coming from item features (such as price). An RS working with AS would need to have domain knowledge to allow more sophisticate evaluation of ASes attributes than simple equality or difference[9].

With these considerations, we concluded that a constraint-based knowledge-based RS is best suited to address our problem.

5.4 Building a constraint-based RS prototype

In order to test our chosen approach, we built a constraint-based RS prototype. This prototype transforms inputs from various information sources (described in section 5.5) into peering recommendations. It does so by applying filter rules that work as our domain constraints, and provide a graphical output of the resulting recommendations.
Our prototype requires a dataset to produce recommendations. The inputs, structure and components of this dataset are outlined in section 5.5 and described in more detail in section 6.2.1.
With the prototype and datasets available, we can start producing recommendations for each scenario in Table 1. In order to assess whether these recommendations are valid and relevant, and the data complete and accurate, we define an evaluation approach in section 5.6. The functionality implemented in the prototype is limited to the minimum required to perform the recommendations. The prototype was written in Python and using open source libraries: Pandas [23], Dash [11].

5.5 Information Sources

From section 5.1 it follows that the attributes required are: network exchange information, peering relationships, traffic information; BGP session configuration and network prefixes per AS. Unfortunately, the availability and quality of data in the public domain is limited[20], and their effects discussed in section 8.1.1.

To obtain exchange information, we used the application programming interface (APIs) of PeeringDB (PDB) [5]. Here, we queried the IXPs where AS1103 is connected to, and retrieved, for each exchange, all available ASes at that IXP. CAIDA's AS Relation dataset[2] was used as source for the peering relationships among ASes. It is publicly available and covers approximately 63.000 ASes. Information regarding volume of traffic was provided by SURFnet's monitoring systems[24]. BGP session configuration and the list of prefixes in the routing tables came from the network infrastructure of SURFnet[25][26]. Finally, the list of network prefixes that belong to an AS were acquired using BGP view APIs[27].

5.6 Evaluating the constraint-based RS prototype

We focus on assessing the degree in which SURFnet's scenarios are fulfilled, as main evaluation approach. Concretely, we performed test on the prototype's Analysis component by processing a test dataset, and verifying manually that the resulting ASes indeed fulfilled the constraints. With the full dataset, we evaluated manually samples of the results, for scenarios 1 and 4. For scenarios 2 and 3, we used the "remarks" field to record the information on why an AS was recommended. To verify the dataset itself, we manually evaluated the data against information coming from SURFnet network infrastructure, and the dataset's input files).

6 Implementation

In the following sections we define how our constraint-based recommender prototype handles its recommendation tasks, and describe its implementation.[1]

6.1 Filter Definitions

We define below the filters required for each of SURFnet's scenarios.

6.1.1 ***Scenario 1****: Propose suitable new Peers* Formulating the filtering rules for scenario 1 allow us to identify the information required. The attributes for this scenario are the list of **peering** and **exchanges** of the ASes.

id	constraint
$filt_1$	the *peerings* attribute of the prospect's AS does not contain SURFnet in any form of peering relationship
$filt_2$	the *exchanges* attribute of the prospect's AS does contain at least one of the IXP where SURFnet is present

Table 2. Filter constraints for scenario 1

6.1.2 ***Scenario 2****: Propose the establishment of BGP sessions if Peer missing in a router* Also here we identify the required attributes: list of **peering**, **exchanges** and **BGP sessions**, per router.

id	constraint
$filt_1$	the *peerings* attribute of the prospect's AS does contain SURFnet as peer
$filt_2$	the *exchanges* attribute of the prospect's AS does contain at least one of the IXP where SURFnet is present
$filt_3$	AS1103's *BGP_session's Remote AS* list does not contain the prospect's AS number, in all *shared_exchanges*

Table 3. Filter constraints for scenario 2

[1] Source code available at: https://gitlab.os3.nl/dgaray/rp1_dashboard

6.1.3 ***Scenario 3****: Propose migrating peering via Route Server to a direct peering form* Here, the difference with scenario 1 is that we additionally need to identify suitable ASes advertising prefixes via route servers. Required attributes for this scenario are the list of **peering**, **exchanges** and ASes' **network prefixes**; and a list with all network prefixes installed on the network infrastructure, along with the AS path information.

id	constraint
$filt_1$	the *peerings* attribute of the prospect's AS does not contain SURFnet as peer
$filt_2$	the *exchanges* attribute of the prospect's AS does contain at least one of the IXP where SURFnet is present
$filt_3$	one or more of the prospect's *prefixes* attribute AS is present in SURFnet's network infrastructure *installed prefixes*
$filt_4$	for each of the prospect's prefix installed on SURFnet's network infrastructure, each individual AS in the *AS Path* does not have any type of peering relationship with SURFnet

Table 4. Filter constraints for scenario 3

6.1.4 ***Scenario 4****: Propose disconnecting peers when traffic is no longer significant* As in scenario 2, we filter here for peers in exchanges where SURFnet is present. Additionally, we evaluate the peak traffic attribute to identify ASes below a specified threshold. Required information here are **peering** lists, and **peak traffic** per AS.

id	constraint
$filt_1$	the *peerings* attribute of the prospect's AS does contain SURFnet as peer
$filt_2$	the prospects *peak traffic* attribute is below a specified threshold *shared_exchanges*

Table 5. Filter constraints for scenario 4

We selected the arbitrary value of 67344739 bits/s as threshold in order to verify the filters (this value was chosen verification and to ensure the result AS set would be non-empty, considering the values in the dataset[24]).

6.2 Prototype Implementation

Considering the specialised functions we identified, we define the following functional components:

- **Data Ingestion and Pre-processing:** which collects and maps information from different sources into our domain model format.
- **Data Analysis**: which performs the recommendation analysis.
- **Visualisation**: providing the recommendation results graphical representation.

These components are depicted in Figure 5.

6.2.1 Domain Model and Dataset definition With the filter definitions and information elicited in sections 6.1.1 through 6.1.4, we define our data model as depicted in Figure 6.

For scenarios 2 and 4, a "remarks" field was generated to specify the reason an AS was included into the resulting set (i.e., addressing in this way the scrutiny requirement). With the definitions above and having mapped the sources of information, we can create a dataset from which we will make our recommendations.

6.2.2 Visualisation The ASes fulfilling the filter criteria are graphically represented in our prototype. In order to compare and rank the results against each other, we use the metrics "peer count" and "prefix count", the number of peers and prefixes an AS has, respectively.

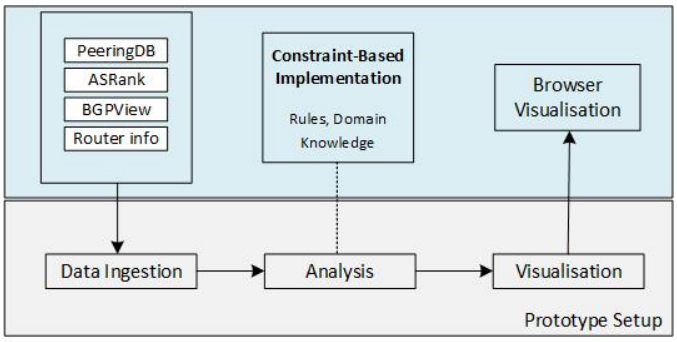

Figure 5. Overview of components and information flows

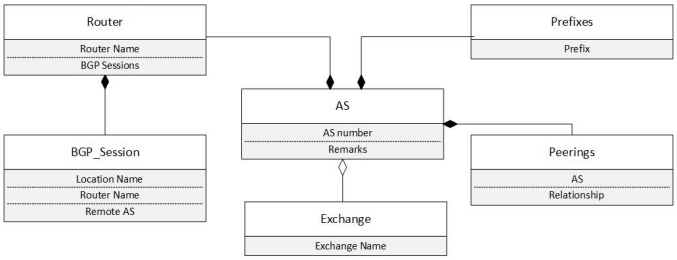

Figure 6. Data Model for the BGP peering domain

7 Results

We present in the following sections the results collected after running our prototype on the data sets.

7.1 Scenario 1: Propose suitable new peers

We analysed 63468 ASes (number of ASes present in CAIDA's AS relationship dataset), of which 284 fulfill the requirement: non-peers available at an exchange where SURFnet is present.

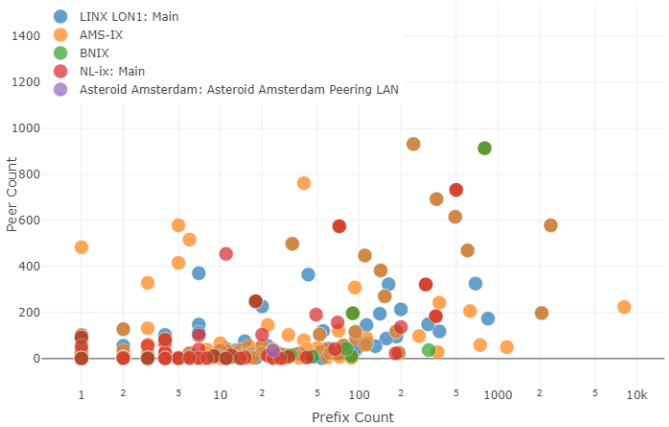

Figure 7. Example output of scenario 1: Propose suitable new peers

As discussed in 6.2.2, ASes currently not peering with SURFnet are depicted in Figure 7 as dots, and ranked according to the number of owned networks and peering relationships ("Prefix Count" and "Peer Count", respectively). For example, a user exploring suitable new peerings in AMS-IX, would look for outliers (larger ASes, according to our criteria) in Figure 7, which belong to AMS-IX.

Example results in the resulting set are: AS109 (Cisco AS)

and AS1257 (Tele2). Note: a logarithmic scale is used on on x-axis of Figure 7.

7.2 Scenario 2: Propose the establishment of BGP sessions if peer missing on a router

We analysed 63468 ASes, of which 980 are peers present in at least one exchange where SURFnet is present. Of these ASes, 15 have BGP sessions with only one of the routers of SURFnet.

An example AS from the resulting set is: AS3267 (Verizon Com). The following remarks were generated for this AS: `Missing session in: Asd001b, exchange: AMS-IX`.

7.3 Scenario 3: Propose migrating traffic handled by Route Servers to a new peer

Here we analysed 63468 ASes, of which 284 are non-peers present in at least one exchange where SURFnet is present. Of these ASes, 282 contain prefixes installed by SURFnet routers (thus fulfill filters filt1, filt2 and filt3 in Table 4). The last verification is to attest if the installed prefix of the AS does not have a peer of SURFnet in the AS path (filt4 in Table 4), which is fulfilled by 2 ASes.

Examples from the resulting set are: AS50384 (W-IX) and AS199522 (Tedas.nl). The ”remarks” field in this case include the installed prefixes found on SURFnet routers, and their corresponding AS paths. For example, for AS50384 the first entry becomes: `79.142.96.0/22:50384:I`.

This means to say, AS50384’s prefix ”79.142.96.0/22” is an installed prefix, and its AS path ”50384:I” (consisting of one AS) does not contain peers of SURFnet.

7.4 Scenario 4: Propose disconnecting peers when traffic is no longer significant

Here we analysed 63468 ASes, of which 980 fulfilled the initial criteria: peers present in at least one exchange where SURFnet is present. Out of this set of ASes, the peak traffic attribute of 74 ASes are below the specified threshold discussed in section 6.1.4.

8 Discussion

In this section we evaluate the prototype execution and results. We reflect on how our research addresses the formulated research questions and finally leave recommendations for future work.

8.1 Prototype Evaluation

We will evaluate the prototype by looking at the following aspects: data ingestion, the analysis and the visualisation components.

8.1.1 Data aspects: there are several sources for possible errors that influence the **accuracy** of our recommendations. For instance, the information about exchanges come from PDB, which is populated on a voluntary basis, so the information available might not be complete and up to date. Scenario 1 might thus recommend less ASes present in a given exchange than there actually are, and perhaps some recommended ASes might have left the exchange in the meantime. Another case is the peering information of ASes, where CAIDA’s AS relationship dataset was used as source. In scenario 2, we identified cases were a peering relationship was reported while there was none, after validating against SURFnet’s network infrastructure information. In scenario 2, we considered the more accurate inputs from SURFnet[25], and therefore the results are accurate. Scenario 3 has a specific filter looking at non-peers in the AS path, and this information can be verified by looking at the ”remarks” field. However, the initial set of ”non-peers” evaluated is smaller due to the inference errors. We suggest that this type of problem will be recurring since many sources of information need to be integrated, and this suggests that there is a need for a structured approach to manage inconsistencies among the data sources. Scenario 3 uses the prefix information coming from BGP View. These prefixes were checked against the installed prefixes on SURFnet’s network infrastructure, and is a reliable source of information. Finally, in scenario 4 we used peak traffic collected from SURFnet’s monitoring systems, which is a reliable source of information.
The **extraction time** of routing table information, BGP sessions and Peak Traffic information, and its processing requires time. For instance, the manual collection of routing table information took approximately 1 hour when extracting from the routers. Another aspect we highlight is the accessibility of systems: in our experience the availability of e.g. APIs to gather information simplified the collection of information.

In general, we suggest that the data ingestion component should account for possible contradictions among data sources, and allow ISPs to manage these conflicts, enriching and annotating the dataset.

8.1.2 Analysis component Once the effects of the data quality and availability are accounted for, the list of ASes returned by the prototype are as expected (as defined in section 5.6), both for scenario 1 and 2.
Further improvements to scenario 3 were discussed, namely to verify the next hop IP address and validate if it belongs to an IXPs IP range. We didn’t find any scenarios that would justify the need for implementing this additional check.

The results of scenario 4 are as expected, within the limitations of the dataset.

The filters used in our prototype were static. We are aware of the prototype’s limitations in e.g. not having more adaptive and dynamic filters, and suggesting or relaxing filtering criteria according to the resulting set of ASes[10].

8.1.3 Visualisation component Ideally, the resulting information should be actionable. For instance, in scenario 2, the results might indicate a configuration problem which can be addressed immediately (unless, of course, it is a design exception, in which case the situation is acceptable). In our prototype, the recommendation information visualised was static, since processing the recommendations dynamically would make, in our opinion, our implementation slow. More sophisticated frameworks for data analysis might allow real time information to be visualised. Although not verified, the recommendation information format based on our domain model can be easily made available via web services API, facilitating the automation of workflows and integration with new data sources.

8.1.4 Research Questions Regarding the methods available for representation and processing of the peering relations, our literature study indicated that graph representations might not be the best approach to answer the practical questions SURFnet formulated. We characterised the different scenarios as an information filtering problem, and identified the constraint-base recommendation system approach as the most suitable to this domain, after evaluating other RS alternatives. Redundancy was explored in scenario 2, where recommendations were made per AS when a BGP session was missing on a router. While other scenarios (e.g. provider multi homing, or route backups) were not included in our study, we suggest that new scenarios exploring redundancy optimisation can also be approached using constraint-based RSs.
Information required and sources: we define in section 5.5 the attributes and sources used in our prototype, necessary to fulfill our required scenarios.

8.1.5 Future Work Apart from the specific suggestions made in the previous section; we suggest the following topics for further research:investigation of other recommendation approaches to improve the ranking of ASes according to other criteria (for instance: traffic, AS Path length, delay, destinations available). Performing calculations at scale and in real time. Evaluate data-processing oriented frameworks, for instance the open-source project PNDA.

9 Conclusion

In this paper we evaluated four concrete BGP peering optimisation scenarios formulated by SURFnet, looked at different filtering techniques and data sources, and finally proposed and built a prototype of a constraint-based Recommendation System. In doing so, we answered our research questions:

Which methods are available for the representation and processing of the peering relations and make optimisation recommendations?
What information and which information sources should be available as input for a tool to fulfill SURFnet's requirements? Can these methods and tools also recommend peers for the best redundancy?

Our approach represents a viable method to generate BGP peering optimisation recommendations, when testing on our collected dataset. We discuss that inaccuracy in the peering information migh result in a smaller set of recommended ASes, and suggest systems for the management of inconsistencies among information sources. We suggest as future work further exploration of BGP peering data, and application of new RS techniques.

References

[1] J. R. Matthew Caesar, "Bgp routing policies in isp networks," 2005.
[2] The Center for Applied Internet Data Analysis (CAIDA), "AS Relationships," [Online]. Available: http://data.caida.org/datasets/as-relationships/. [Accessed: January 10, 2019].
[3] IETF, "A Border Gateway Protocol 4 (BGP-4)," https://tools.ietf.org/html/rfc4271 (Accessed on 2019-1-10).
[4] The Center for Applied Internet Data Analysis (CAIDA), "Center for Applied Internet Data Analysis," [Online]. Available: http://www.caida.org/home/. [Accessed: January 8, 2019].
[5] PeeringDB, "PeeringDB," [Online]. Available: https://www.peeringdb.com/. [Accessed: January 8, 2019].
[6] X. D. Pavlos Sermpezis, George Nomikos, "Re-mapping the internet: Bring the ixps into play," 2017.
[7] M. E. Toza, "Policy-preferred paths in as-level internet topology graphs," in *Theory and Applications of Graphs: Vol. 5: Iss. 1 , Article 3.*
[8] M. E. T. Abdullah Yasin Nur, "Cross-as (x-as) internet topology mapping," in *Computer Networks 132 (2018).* School of Computing and Informatics, University of Louisiana at Lafayette, 2017, pp. 53–67.
[9] M. R. Robin Burke, "Matching recommendation technologies and domains," in *Recommender Systems Handbook*. IEEE, 2010, pp. 367–382.
[10] R. B. A. Felfernig, "Constraint-based recommender systems: Technologies and research issues," 2015.
[11] Plotly, "Dash - Build beautiful web-based interfaces in Python," [Online]. Available: https://dash.plot.ly/. [Accessed: December 28, 2018].
[12] PNDA Project, "The scalable, open source big data analytics platform for networks and services." [Online]. Available: http://pnda.io/. [Accessed: January 10, 2019].
[13] SURFnet Blog, "Innovatieblog - SURFnet-netwerk Dashboard ," [Online]. Available: https://blog.surf.nl/tag/surfnet-netwerk-dashboard/. [Accessed: December 28, 2018].
[14] Jac Kloots, "Hoe bieden we onze instellingen een optimale internetverbinding?" [Online]. Available: https://blog.surf.nl/hoe-bieden-we-onze-instellingen-een-optimale-internetverbinding/. [Accessed: January 23, 2019].
[15] AMS-IX, "AMS-IX Amsterdam," [Online]. Available: https://www.ams-ix.net/ams. [Accessed: January 21, 2019].
[16] BNIX, "BNIX," [Online]. Available: https://www.bnix.net/nl. [Accessed: January 21, 2019].
[17] LINX, "The London Internet Exchange," [Online]. Available: https://www.linx.net/. [Accessed: January 21, 2019].
[18] NL-IX, "NLix The Interconnect Exchange," [Online]. Available: https://www.nl-ix.net/. [Accessed: January 21, 2019].
[19] Amsterdam Asteroid, "Asteroid IXP," [Online]. Available: https://www.asteroidhq.com/. [Accessed: January 21, 2019].
[20] A. D. C. D. Aemen Lodhi, Nikolaos Laoutaris, "Complexities in internet peering: Understanding the black in the black art," 2015.
[21] AMS-IX, "AMS-IX Route Servers," [Online]. Available: https://www.ams-ix.net/ams/documentation/ams-ix-route-servers. [Accessed: January 21, 2019].
[22] B. S. P. B. K. Francesco Ricci, Lior Rokach, *Recommender Systems Handbook*. Springer, 2010.
[23] Pandas, "Pandas - Python Data Analysis Library," [Online]. Available: https://pandas.pydata.org/. [Accessed: December 28, 2018].
[24] SURFnet, "Top 350 peer," [Online]. Available: https://gitlab.os3.nl/dgaray/rp1_dashboard/tree/master/proto_data_ingestion/Resources/peak_traffic. [Accessed: January 28, 2019].
[25] ——, "Router information - showbgpsum," [Online]. Available: https://gitlab.os3.nl/dgaray/rp1_dashboard/tree/master/proto_data_ingestion/Resources/sessions. [Accessed: January 28, 2019].
[26] ——, "Router information - Installed prefixes," [Online]. Available: https://gitlab.os3.nl/dgaray/rp1_dashboard/tree/master/proto_recommender/Resources/surfnet_rt. [Accessed: January 28, 2019].
[27] bgpview.io, "BGP View," [Online]. Available: https://bgpview.io/. [Accessed: January 21, 2019].