

# Extraheren en voorspellen van trafiekmatrices als input voor planning van telecommunicatienetwerken

Axl François

Promotoren: prof. dr. ir. Sofie Verbrugge, prof. dr. ir. Didier Colle

Begeleiders: Jonathan Spruytte, mevr. Elise Vivier (Proximus)

Masterproef ingediend tot het behalen van de academische graad van  
Master of Science in de industriële wetenschappen: informatica

Vakgroep Informatietechnologie  
Voorzitter: prof. dr. ir. Bart Dhoedt  
Faculteit Ingenieurswetenschappen en Architectuur  
Academiejaar 2016-2017





**Vertrouwelijk tot en met 30/06/2022**

**Belangrijk**

Deze masterproef bevat vertrouwelijke informatie en/of vertrouwelijke onderzoeksresultaten die toebehoren aan de Universiteit Gent of aan derden. Deze masterproef of enig onderdeel ervan mag op geen enkele wijze publiek gemaakt worden zonder de uitdrukkelijke schriftelijke voorafgaande toestemming vanwege de Universiteit Gent. Zo mag de masterproef onder geen voorwaarde door derden worden ingekeken of aan derden worden meegedeeld. Het is verboden om de masterproef te kopiëren of op eender welke manier te dupliceren. Indien de vertrouwelijke aard van de masterproef niet wordt gerespecteerd, kan dit onherstelbare schade veroorzaken aan de Universiteit Gent. Bovenstaande bepalingen zijn van kracht tot en met de embargodatum.

# Extraheren en voorspellen van trafiekmatrices als input voor planning van telecommunicatienetwerken

Axl François

Promotoren: prof. dr. ir. Sofie Verbrugge, prof. dr. ir. Didier Colle  
Begeleiders: Jonathan Spruytte, mevr. Elise Vivier (Proximus)

Masterproef ingediend tot het behalen van de academische graad van  
Master of Science in de industriële wetenschappen: informatica

Vakgroep Informatietechnologie  
Voorzitter: prof. dr. ir. Bart Dhoedt  
Faculteit Ingenieurswetenschappen en Architectuur  
Academiejaar 2016-2017



# Voorwoord

Deze masterproef werd geschreven met het oog op het behalen van het diploma Master Of Science in de industriële wetenschappen: informatica. Het zou onmogelijke geweest om deze masterproef te schrijven zonder de hulp van enkele mensen. Ik wil dan ook een woord van dank tot hen richten.

In eerste plaats wil ik Jonathan Spruytte voor de constructieve meetings en feedback gedurende de volledige masterproef bedanken. Door elke meeting krijg ik nieuwe ideeën en motivatie om verder te werken aan de masterproef.

In het algemeen wil ik ook Proximus bedanken voor het aanleveren van de data die deze masterproef mogelijk maakt. Specifiek zou ik Elise Vivier willen bedanken voor haar deskundig inzicht over het Proximus netwerk. Maar ook voor het advies en feedback die ik gedurende de masterproef gekregen heb.

Ik wil ook nog de promotoren prof. dr. ir. Soffie Verbrugge en prof. dr. ir. Didier Colle van deze masterproef bedanken voor hun feedback.

Ten slotte wens ik ook nog mijn ouders en vrienden bedanken voor de steun tijdens het maken van deze masterproef.

Axl François, 12 juni 2017

# Toelating tot bruikleen

De auteur(s) geeft (geven) de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.

The author(s) gives (give) permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In the case of any other use, the copyright terms have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master dissertation.

Axl François, 12 juni 2017

# Abstract

Elk jaar zijn er nieuwe technologieën beschikbaar die ons in staat stellen beter met elkaar te communiceren. Deze technologieën leggen eisen op aan de onderliggende telecommunicatienetwerken. 5G, 4K-streaming, . . . zijn voorbeelden hiervan. Elk van deze technologieën eisen meer bandbreedte dan hun voorgangers.

Het is dus aan telecommunicatie-operatoren om met een beperkt budget een zo kwalitatief mogelijke dienst aan te bieden. De netwerkinfrastructuur van deze operatoren moet voldoen aan de eisen die gesteld worden door deze nieuwe technologieën. Een upgradeplanning die rekening houdt met de huidige vereisten maar ook met de toekomstige, is een *must*.

In deze masterproef worden twee analyses geïntroduceerd die helpen bij het opstellen van de upgradeplanning. Op basis van metingen binnen het Proximus-netwerk worden trafiek matrices opgesteld die daarna gesimuleerd en geëvalueerd worden. Dezelfde metingen kunnen gebruikt worden om meer inzicht te krijgen in hoe trafiek zich gedraagt binnen het Proximus-netwerk.

Het resultaat van deze masterproef stelt Proximus in staat om een beter upgradeplanning te maken en meer inzicht te krijgen in hun netwerk. Het programma dat geschreven is, gedurende deze masterproef zorgt er voor dat Proximus op een gemakkelijk manier informatie kan verzamelen. Deze informatie kan later gebruikt worden om de upgradeplanning verder te verbeteren.

# Abstract

Each year new technological advancements are made. These advances help to facilitate communications. They also impose requirements on the underlying telecommunication networks. Technologies like 5G, 4K-streaming, . . . each require more bandwidth than their predecessors.

Telecommunication-operators face the challenge of having a restricted budget to upgrade the current infrastructure to meet the demands imposed by these technologies and services. An upgrade planning that takes into account not only the current situation but also the future demands, is a must.

In this article two analysis are introduced to help improve the upgrade planning. First the current demands in the Proximus-network are used to build traffic matrices which are then simulated and evaluated. Secondly traffic measurements are used to gain more insight in how the traffic behaves within the Proximus-network.

The results put forward in this article allow Proximus to better plan network-upgrades and have more insight into their network. The tool that was created in this process, allows Proximus to easily build up a wealth of information. This information can later be exploited to further improve the upgrade planning.



# Extraction and prediction of traffic matrices as input for planning of telecommunication networks

Axl François

Promotor(s): prof. dr. ir. Soffie Verbrugge, prof. dr. ir. Didier Colle

Supervisor(s): Jonathan Spruytte (UGent), Elise Vivier (Proximus)

*Abstract*—Telecommunication-operators face the challenge of having a restricted budget to upgrade the current infrastructure to meet ever increasing demands. Measurements from the Proximus-network are used to improve the upgrade planning. Several aggregation levels are introduced. These are used to build up traffic matrices and trends. Traffic matrices are used to simulate networks and investigate the impact of certain demands. The trends help to give more information about how the traffic behaves within the Proximus-network, specifically the daily and monthly pattern is researched for internet services. The daily pattern is also used to make heatmaps, that show the evolution of a service throughout the day.

*Keywords*—traffic matrices, upgrade planning

## I. INTRODUCTION

Each year new technological advancements are made. These advances help to facilitate communications. They also impose requirements on the underlying telecommunication networks. Technologies like 5G, 4K-streaming, ... each require more bandwidth than their predecessors.

Telecommunication-operators face the challenge of having a restricted budget to upgrade the current infrastructure to meet the demands imposed by these technologies and services. An upgrade planning that takes into account not only the current situation but also the future demands, is a must.

The first step towards improving the upgrade planning is to understand how demands behave within a network. Therefore it is very useful to make measurements of the throughput at each node. Not every node will have the same amount of customers and not every customer will have the same demands. These measurements can be used to clarify the differences in demands, between nodes and customers. These measurements can also be correlated with the location of these nodes, to see if there is any geographical differences.

A second step, is an extensive analysis of what-if scenarios. In these scenarios important questions are asked, like what if a node fails? What if the traffic of a link in the network increases? What if a node or region gets more customers? In these hypothetical situations it is important to know if the network can survive and meet the demands. If not than this should be taken into account for the upgrade planning.

In this thesis the measurements are provided by Proximus. A tool was written to aggregate and analyze these measurements. The goal is to provide usefull insights that can be used in the upgrade-planning. The structure of the paper is as following: in section II the available data and its structure is given. Next

in section III the tool to analyse and aggregate the data is discussed. Furthermore in section IV traffic matrices are build using the output made in section III. Finally in section V the different trends like the daily and monthly pattern are constructed.

## II. DATA

The Proximus-network has  $\approx 600$  nodes. For each of these nodes measurements are provided. These measurements happen continuously and are averaged over five minutes for both ingress and egress. The nodes can be divided over core, distribution and access layers. Each access node connects to a distribution node and each distribution node connects to a core node. Each access and distribution node has at least one ethernet service switch (ESS) which connects to a regional node.

The measurements can further be divided into two groups namely aggregator and service measurements. The aggregator measurements are bound by the technology that is used to aggregate customers onto the Proximus-backbone network. One ESS can have multiple aggregators. It are these aggregators that are being measured. In figure 1 a network overview of this is given.

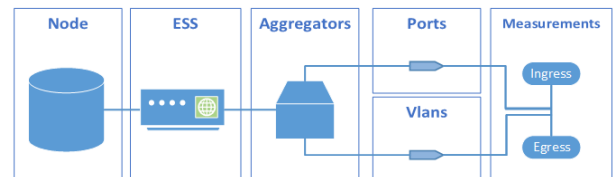


Fig. 1. Overview of where the measurements are being taken within the Proximus network hierarchy.

The service measurements are made per vlan. Each vlan has an unique identifier. A mapping between vlans and services is made in the tool. Each aggregator handles traffic for multiple vlans thus the service measurements are a split up of the aggregator measurements per vlan.

It is also important to note that the measurements reflect the local demands of each node. In figure 2, a small example of this is shown. Both node A en B connect to the regional node C. The measurements of node C do not include the demands of Node A or B.

Furthermore information on customer numbers has also been provided. This information is also split up between aggregators

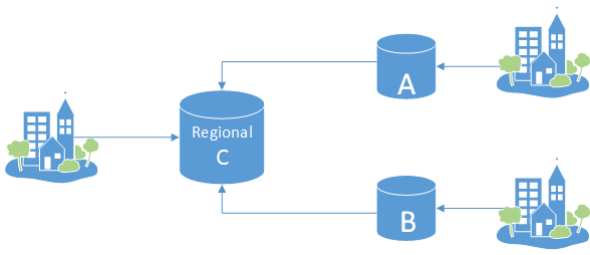


Fig. 2. A small network example with three nodes and local demands (shown as cities).

and services.

### III. TOOL

A tool was developed that allowed for the data to be parsed, aggregated and analyzed. This tool is written in Java and has several configurable components. A flexible object model was introduced to meet future requirements. In figure 3 an overview is given of the workflow of the tool.

The measurements are divided over several files, each of these files can have a different format. The different technologies being measured were already discussed in section II. A flexible parsing environment was introduced. This was achieved by introducing a configuration file per type of input file. A general configuration file was also introduced for the fields that stay the same over all the input files, like the node identification. The result of parsing an input file is a list of domain objects.

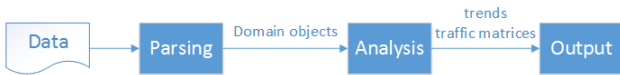


Fig. 3. General workflow of the tool to process the data from Proximus.

In the analysis phase of the tool, these domain objects are used to make aggregations which are then used to make traffic matrices (section IV.) and trends (section V.). Several aggregation levels are introduced corresponding with the network hierarchy. In figure 4 these aggregation levels are shown, note that these aggregations can also be made for both aggregators and services.

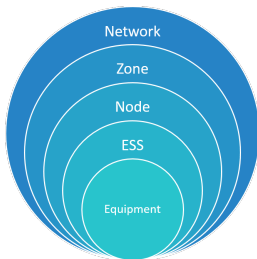


Fig. 4. An overview of all the possible aggregation levels in the tool.

### IV. TRAFFIC MATRICES

A traffic matrix is an  $n \times n$ -matrix with  $n$  equal to the number of nodes in the network. Each value in the matrix represents the throughput at a given time from node  $i$  to node  $j$ . Note that an  $n \times n$ -matrix implies that every node communicates with every other node in the network, for the Proximus network this is not the case. The output of the tool is used as input for making traffic matrices, specifically the demands per service per node.

As shown in figures 1 and 2, the measurements do not give us any information about the origin of the demands. Thus the concept of injection points is introduced. These are initially virtual nodes that act as the source of these demands. In other words these injection points fulfill the demands of the nodes within the network. For the demands to be met, at least one injection point is needed. The question is then asked what happens with the network if more injection points are introduced and if the demands are divided over these injection points. Two variables are noted here, the number of injection points and the way the demands should be divided over these injection points.

In the simplest case one injection point is responsible for all the demands in the network, this would result in a  $1 \times 600$  matrix for the Proximus network. Once a second injection point is added, the question becomes how should the demands be divided over the injection points?

An option is to look for all possible permutations of these traffic matrices, given a number of injection points. Note that this is a permutation with repetition. For each node an injection point has to be chosen. For every node the same injection points can be chosen, hence repetition. It is a permutation because the order in which the injection points are chosen is important.

So for two injection points there are  $2^{600}$  possible matrix permutations. The result of such a permutation is not a valid traffic matrix, the injection points still have to be mapped on real nodes. Since every unique combination of two nodes is valid, there are  $\binom{600}{2}$  possibilities. The total number of traffic matrices that would have to be simulated to find an optimal solution is  $2^{600} * \binom{600}{2}$ . This is infeasible.

The number of possibilities should be reduced before attempting to find a suboptimal solution for this problem. A possible approach is to take into account the hop count when assigning injection points. If for each node the injection point is chosen with the lowest hop count than there will be less traffic in the network. Another option is to limit the amount of possible injection points.

Once a valid traffic matrix is made, it is possible to simulate the network based on this traffic matrix and calculate the needed capacities on the links of the network. This happens using a network simulation library. Multiple traffic matrices can be generated and then simulated to research their impact on the network. By carefully choosing the injection points it is also possible to reduce the effects of failure. Finding traffic matrices that reduce the effects of failure and the overall link-load is the main goal.

## V. ANALYSIS OF TRENDS

To better understand how traffic behaves within the Proximus network, several analysis are made. One of the questions is, if the demands behave in a predictable pattern or in a more random pattern. For the upgrade planning the peak-throughput is of great importance. In [1] it is shown that distinct diurnal patterns can be observed. It is also noted that spikes in traffic might occur due to ‘unusual traffic behaviour’ such as DDoS attacks, flash crowds, . . . .

The monthly and daily patterns are made and compared between FIA\_RES and FIA\_PRO, these are two services that represent internet access for RESidential and PROfesional users. First the monthly pattern is discussed for March. For each day the peak-throughput is determined and plotted in a graph. The peak-times are also plotted. A peak-time is the moment when peak-throughput happens.

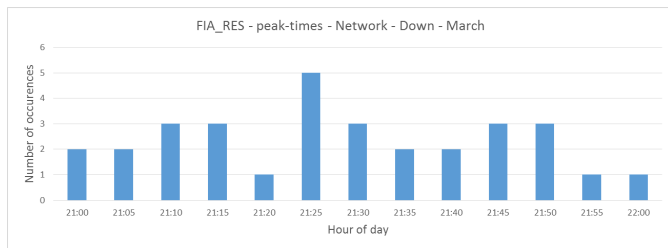


Fig. 5. The peak-times for FIA\_RES in March, down.

For FIA\_RES there are clear lows each Friday and Saturday. The peak-throughput mostly occurs on Sundays. There is still some variations and the weekdays do not show a clear pattern. The peak-times occur between 21:00 - 22:00 hours, see figure 5. FIA\_PRO on the other hand shows a very clear pattern. The weekdays have about the same peak-throughput with one exception. The weekends form clear lows in this graph. The peak-times mostly occur between 9:15 and 12:30, see figure 6.

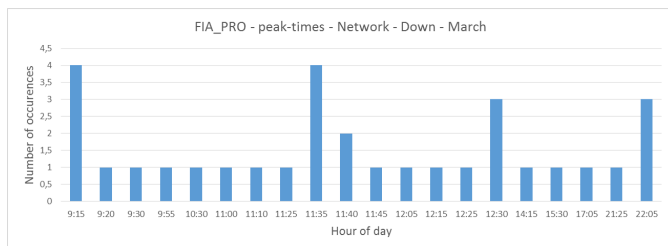


Fig. 6. The peak-times for FIA\_PRO in March, down.

The traffic is then divided over five *IP-traffic aggregation* zones, each zone represents a subset of nodes from the whole network. The patterns of the zones are then compared with each other. For both services zones 2, 4 and 5 behave in roughly the same way. While zone 1 clearly has to process less traffic than the other zones and zone 3 processes more than the other zones.

The daily pattern for FIA\_RES and FIA\_PRO is analyzed and several trends are noted. These trends are split-up in different graphs. FIA\_RES shows three different trends namely Wednesdays, the other weekdays, weekends<sup>1</sup>. FIA\_PRO has two distinct trends, weekdays and weekends.

The contribution per customer is calculated and compared between FIA\_RES and FIA\_PRO. This is shown for ISAM\_HOSTS<sup>2</sup>. FIA\_PRO customers use more traffic in the weekdays than FIA\_RES customers.

Finally the daily pattern per service is used to make heatmaps. These heatmaps show the evolution of traffic throughout the day. The throughput is scaled and divided over several groups. Every five minutes and image-file is made, these image files are put together into a GIF-animation. This animation clearly shows the evolution of the daily pattern.

## VI. FUTURE WORK

The future of the research is located in two areas namely the prediction of traffic demands and an extensive study on how to solve the optimization problem put forward in section IV. using traffic matrices. As an extra several extensions can be written to the tool developed in section III.

The first step towards the prediction of traffic demands was made in this article. Proximus should use the tool to create a historic data trove. Several prediction models should be selected based on the characteristics of the data (time series, . . .). A comparative study of these prediction models should be made using the data.

In this article a few suggestions have been made to reduce the number of possible permutation when generating traffic matrices. However further research is still needed to find a (sub)optimal solution for this problem.

The tool (section III.) can also be improved. Better user interaction in the form of a graphical user interface is a possibility or perhaps a more advanced command line interface.

## VII. CONCLUSION

The tool that was created in this process, allows Proximus to easily build up a wealth of information. This information can later be exploited to further improve the upgrade planning.

A methodology was introduced to build up traffic matrices using injection points. Networks can be simulated and evaluated based on these traffic matrices. The overall link-load can be reduced by carefully choosing the injection points.

Several trends are noted for FIA\_RES and FIA\_PRO. These trends provide Proximus with a better insight of these services. Although only two services are discussed in this article, these analysis can also be made for every other combination of vlans.

<sup>1</sup>Weekend: Saturday and Sunday

<sup>2</sup>This is a type of aggregator.

The results put forward in this article allow Proximus to better plan network-upgrades and have more overall insight into their network.

#### REFERENCES

- [1] Paul Tune and Matthew Roughan, "Internet Traffic Matrices : A Primer," in *Recent Advances in Networking*. 2013.

# Inhoudsopgave

Inhoudsopgave	xiii
Lijst van figuren	xv
Lijst van tabellen	xvii
List of terms	xviii
<b>1 Inleiding</b>	<b>1</b>
1.1 Meerlagige netwerken	3
1.2 Falen van netwerken	5
1.3 Trafiekmatrices	6
1.4 Onderzoeksproject	8
<b>2 Data</b>	<b>11</b>
2.1 Metingen per aggregator	12
2.1.1 Bedraad	12
2.1.2 Klanten informatie	13
2.1.3 Draadloos	13
2.2 Metingen per service	14
2.2.1 Bedraad	14
2.2.2 Draadloos	15
2.2.3 Klanten informatie	15
2.3 Geografische informatie	16
2.4 Voorkomende problemen	16
<b>3 Tool</b>	<b>18</b>
3.1 Parsen	18
3.1.1 Configuratiebestanden	19
3.2 Aggregaties	20
3.3 Aggregatie problemen	21

---

3.3.1	Klanten . . . . .	22
3.3.2	Voorbeeld van een mogelijke applicatieflow . . . . .	23
3.4	Gebruik . . . . .	25
3.5	Uitbreidbaarheid . . . . .	26
<b>4</b>	<b>Opbouwen van Trafiekmatrices</b>	<b>27</b>
4.1	Theoretisch Voorbeeld . . . . .	28
4.2	Simulaties . . . . .	32
4.3	Toepassing . . . . .	34
4.4	Conclusie . . . . .	35
<b>5</b>	<b>Analyse van trends</b>	<b>36</b>
5.1	Analyseren van trafiek . . . . .	37
5.1.1	Dienst FIA_RES . . . . .	37
5.1.2	Dienst FIA_PRO . . . . .	41
5.2	Analyseren van <i>daily pattern</i> . . . . .	44
5.2.1	Analyseren van <i>daily pattern</i> voor FIA_RES . . . . .	44
5.2.2	Analyseren van <i>daily pattern</i> voor FIA_PRO . . . . .	47
5.3	Contributie per klant . . . . .	49
5.4	Heatmaps . . . . .	49
5.5	Conclusie . . . . .	52
<b>6</b>	<b>Conclusie</b>	<b>53</b>
6.1	Toekomstige onderzoeksperspectieven . . . . .	54
6.1.1	Uitbreidingen aan de tool . . . . .	55
6.1.2	Suggesties tot het verbeteren van de inputdata . . . . .	55
6.1.3	Optimale oplossing voor trafiekmatrices . . . . .	56
6.1.4	Trafiek voorspellen . . . . .	56
	<b>Bibliografie</b>	<b>57</b>
	<b>A Permutaties</b>	<b>59</b>

## Lijst van figuren

1.1	Evolutie van bandbreedte voor bedrade en draadloze netwerken [10, 6] . . .	2
1.2	Verdeling van de bandbreedte volgens applicaties/services [12, 4]. . . . .	3
1.3	Een voorbeeld van een meerlagige netwerk [13]. . . . .	5
1.4	Een voorbeeld van een trafiekmatrix [14] . . . . .	7
2.1	Proximus netwerkhiërarchie op basis van een ISAM-host. . . . .	12
2.2	Overzicht van metingen per <i>aggregator</i> binnen de Proximus netwerkhiërarchie	13
2.3	Overzicht van metingen per aggregator per service binnen de Proximus net- werkhiërarchie . . . . .	15
3.1	Workflow van applicatie voor het verwerken van data. . . . .	18
3.2	Een overzicht van mogelijke aggregaties. . . . .	21
3.3	Metingen over verschillende poorten voor hetzelfde tijdstip. . . . .	22
3.4	Aggregatie maken over een groter tijdsinterval dan de beschikbare vijf minuten.	22
3.5	Workflow van applicatie voor het maken van een per CO per dienst per klant overzicht. . . . .	24
4.1	Een netwerk dat als basis dient voor een aantal voorbeeld simulaties. . . .	28
4.2	Voorbeeld simulatie met trafie matrix 4.2 . . . . .	29
4.3	Simulatie met Node B als <i>injection point</i> voor alle demands zie tabel 4.3. .	30
4.4	Simulatie op basis van tabel 4.4 . . . . .	30
4.5	Simulatie op basis van tabel 4.5 . . . . .	31
4.6	Simulatie op basis van tabel 4.6 . . . . .	32
4.7	Simulatie op basis van tabel 4.7 . . . . .	33
4.8	De applicationflow voor het generen van trafiekmatrices . . . . .	35
5.1	De peak-throughput voor de dienst FIA_RES in maart. . . . .	38
5.2	De peak-times voor de dienst FIA_RES in maart. . . . .	38

---

5.3	De <i>peak-throughput</i> verdeeld over de verschillende SIMBA-zones voor de dienst FIA_RES in maart. . . . .	39
5.4	De peak-times van de verschillende zones voor de dienst FIA_RES in maart. . . . .	40
5.5	De <i>peak-throughput</i> bij een aantal nodes van de 02 phonezone voor de dienst FIA_RES in maart. . . . .	40
5.6	De <i>peak-throughput</i> bij de verschillende ESS'en van de 02and-node voor de dienst FIA_RES in maart. . . . .	41
5.7	De <i>peak-throughput</i> voor de dienst FIA_PRO in maart. . . . .	42
5.8	De peak-times voor de dienst FIA_PRO in maart. . . . .	42
5.9	De <i>peak-throughput</i> verdeeld over de verschillende SIMBA-zones voor de dienst FIA_PRO in maart. . . . .	43
5.10	De peak-times van de verschillende zones voor de dienst FIA_PRO in maart. . . . .	43
5.11	De <i>peak-throughput</i> bij een aantal nodes van de 02-phonezone voor de dienst FIA_PRO in maart. . . . .	44
5.12	Het <i>daily pattern</i> voor de dienst FIA_RES in maart. . . . .	45
5.13	Het <i>daily pattern</i> voor week 10 van de dienst FIA_RES in maart. . . . .	45
5.14	Het <i>daily pattern</i> voor alle woensdagen van de dienst FIA_RES in maart. . . . .	46
5.15	Het <i>daily pattern</i> voor alle weekends van de dienst FIA_RES in maart. . . . .	46
5.16	Het <i>daily pattern</i> voor alle weekdays van de dienst FIA_RES in maart. . . . .	47
5.17	Het <i>daily pattern</i> voor alle weekdays van de dienst FIA_RES in maart. . . . .	47
5.18	Het <i>daily pattern</i> voor de dienst FIA_PRO in maart. . . . .	48
5.19	Het <i>daily pattern</i> van een typisch week voor de dienst FIA_PRO in maart. . . . .	48
5.20	Het <i>daily pattern</i> voor alle weekends van de dienst FIA_PRO in maart. . . . .	48
5.21	De peak-throughput contribution per customer voor zowel FIA_RES en FIA_PRO bij ISAM_HOSTS. . . . .	49
5.22	Het <i>daily pattern</i> van de dienst FIA_RES op 01/03. . . . .	50
5.23	Heatmap voor de dienst FIA_RES om 05:10 uur op 01/03/2017 . . . . .	51
5.24	Heatmap voor de dienst FIA_RES om 11:50 uur op 01/03/2017 . . . . .	51
5.25	Heatmap voor de dienst FIA_RES om 21:55 uur op 01/03/2017 . . . . .	52



# Lijst van tabellen

2.1	Een ingekort voorbeeld van een lijn uit de <i>aggregator</i> bestanden. . . . .	12
2.2	Overzicht van alle bedrade <i>aggregator</i> types. . . . .	14
2.3	Een ingekort voorbeeld van een lijn uit de draadloze <i>aggregator</i> bestanden.	14
2.4	Een ingekort voorbeeld van een lijn uit de diensten bestanden. . . . .	15
2.5	Een ingekort voorbeeld van een lijn uit de draadloze diensten bestanden. .	15
2.6	Overzicht van de beschikbare klanteninformatie. . . . .	16
3.1	Een voorbeeld van de inputmanipulatie bij aggregator klanten. . . . .	23
3.2	De gemiddelde uitvoeringstijd voor enkele aggregaties per inputbestand . .	25
4.1	Egress Demands . . . . .	29
4.2	Trafiekmatrix met A als <i>injection point</i> . . . . .	29
4.3	Trafiekmatrix met B als <i>injection point</i> . . . . .	30
4.4	Trafiekmatrix met twee <i>injection points</i> afwisselend toegewezen . . . . .	30
4.5	Trafiekmatrix met twee <i>injection points</i> zonder belasting voor Node A. . .	31
4.6	Trafiekmatrix waarvan de <i>injection points</i> een grote hop-afstand hebben tegenover de vereisten van dienst X. . . . .	32
4.7	Trafiekmatrix met de kleinste link belasting. . . . .	33
4.8	De trafiekmatrix waarvan bepaalde mappings van injection points zorgen voor maximale belasting. . . . .	34
5.1	Gemiddelde <i>peak-throughput</i> in maart voor FIA_RES, down. . . . .	39
5.2	Gemiddelde <i>peak-throughput</i> in maart voor FIA_PRO, down. . . . .	43
A.1	Permutaties voor twee injection points op basis van het netwerk in figuur 4.1	59
A.1	Permutaties voor twee injection points op basis van het netwerk in figuur 4.1	60

# Afkortingën

**4K-streaming** Streaming met 3840 pixels breed en 2160 pixels hoog.

**AGW** Access Gateway.

**CDN** content delivery network.

**ESS** Ethernet Service Switch.

**FTTH** Fiber To The Home.

**GELT** Gigabit Ethernet Line Termination.

**GENT** Gigabit Ethernet Network Termination.

**IEEE** Institute of Electrical and Electronics Engineers.

**IP** Internetprotocol.

**ISAM** Intelligent Services Access Manager.

**LEX** Local Exchange.

**OTN** Optical Transport Network.

**PB** Petabyte -  $10^{15}$  byte.

**POTS** Plain Old Telephone Service.

**RAM** Intelligent Services Access Manager.

**SDH** Synchronous Digital Hierarchy.

**SLA's** Service Level Agreement.

**SONET** Synchronous Optical Networking.

**TCP/IP** Transmission Control Protocol (TCP) en het internetprotocol (IP).

**Ultra HD** Synoniem voor 4k..

**VLAN** Virtual Local Area Network.

**VoD** Video on demand.

**VoIP** Voice over IP.

**WDM** wavelength-division multiplexing.

# Hoofdstuk 1

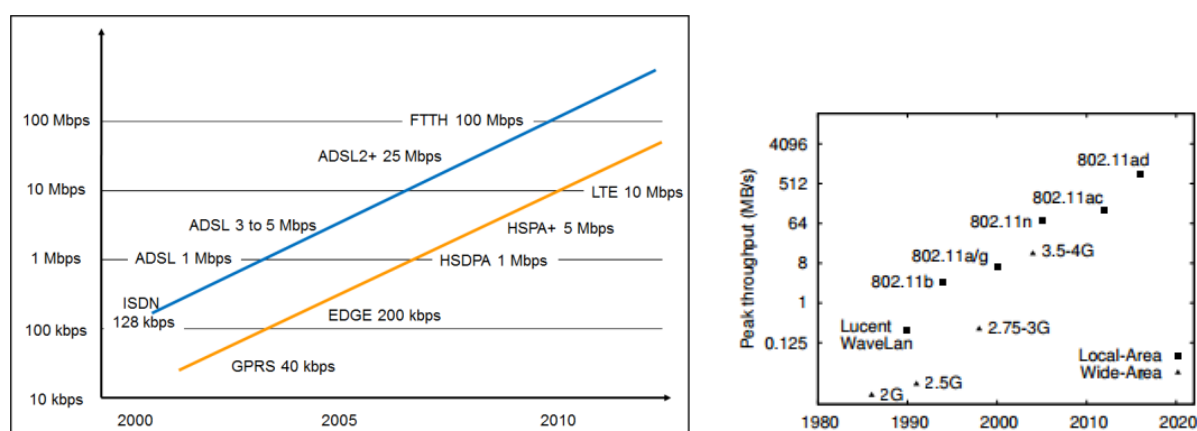
## Inleiding

Elk jaar zijn er nieuwe technologieën beschikbaar die ons in staat stellen beter met elkaar te communiceren. Deze communicatie wordt gedreven door vooruitgang in verschillende gebieden. Nieuwe telecommunicatiestandaarden zoals 5G zorgen, er voor dat we meer data dan ooit kunnen versturen. Daarnaast zijn *video on demand services* (Netflix, ...) steeds populairder en deze eisen voldoende bandbreedte om kwalitatieve streaming te kunnen aanbieden. Specifiek zal 4K-streaming in de toekomst de norm zijn. Het internet zelf groeit ook nog elke dag en cloud services zijn niet meer weg te denken. Zowel de cloud storage services (Dropbox, Google Drive, One Drive) als andere gedistribueerde applicaties zorgen voor meer datatrafiëk (Facebook, Youtube, ...).

De vraag is dan ook hoe we kunnen omgaan met deze altijd toenemende eisen voor de onderliggende telecommunicatienetwerken. Dit zowel op gebied van bandbreedte, uptime en latency. Er zal dus op regelmatige basis moeten geïnvesteerd worden in upgrades voor deze netwerken om de technologieën van morgen te kunnen ondersteunen. Bovendien heeft niet elke datastroom dezelfde prioriteit of hetzelfde patroon in een meerlagige netwerkarchitectuur. Een telecomoperator zal dan ook met een beperkt budget een zo goed mogelijk services willen bieden aan zijn klanten.

Die klanten gebruiken verschillende manieren om toegang te krijgen tot het internet. De voorbije 10-20 jaar zijn deze technologieën enorm geëvolueerd en is er alvast een opmerkelijke trend (Figuur 1.1). Voor zowel mobiel internet als vaste lijnen is er steeds een verhoging van de maximale bandbreedten. De opkomende technologieën die elk weer meer bandbreedte zullen aanbieden dan hun voorgangers zijn: FTTH (Google Fiber: 1 Gbit/s [16]), 5G (tegen 2020: snelheid +7 Gbit/s [5]), IEEE 802.11ad (6.75 Gbit/s voor afstanden

< 5m [11]).



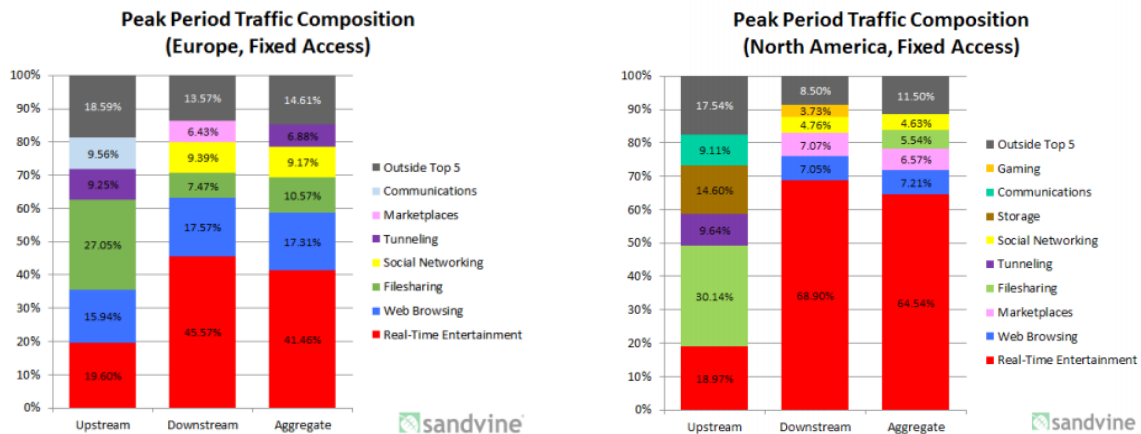
**Figuur 1.1:** Evolutie van bandbreedte voor bedrade en draadloze netwerken [10, 6]

Volgens [12] is 64.54% van al het vast-internet verkeer in Noord-Amerika, realtime-entertainment. Het is dus heel duidelijk dat services zoals Netflix, Spotify en andere content providers een groot deel van het internet verkeer innemen. In figuur 1.3 is er een verdeling te vinden van deze diensten.

Op de website van Netflix stellen ze vast dat er minstens 25 Mbit/s [18] nodig is om Ultra HD te kunnen streamen. Natuurlijk heeft niet iedereen een Ultra HD beeldscherm, maar het is wel een cijfer om rekening mee te houden. Als iedere video service namelijk 4k/Ultra HD content wil aanbieden dan zijn de gevolgen voor de bandbreedte vereisten en de bijhorende congestie op netwerken van operatoren nefast.

Cisco geeft elk jaar een voorspelling voor het verwachte globale internetverkeer [17]. De voorspelling voor 2020 is 194.374 PB per maand. Voor 2017 houden ze het totaal aantal PB per maand op 108.533. Cisco verwacht dus dat het globale internet verkeer zal toenemen met een factor 1.8 binnen 3 jaar. Het grootste deel van dit verkeer zal volgens Cisco naar 'Internet Video' gaan. Een goede content distribution is dus een noodzaak voor telecomoperatoren.

Een analyse van de huidige trafiek in een netwerk kan veel nuttige inzichten aanbieden. Dergelijke analyse stelt een operator in staat om zijn budget beter te benutten bij het maken van de upgradeplanning. In een ideaal scenario weet een operator het toekomstige dataverkeer en bijkomende eisen zodanig dat hij zich hier optimaal kan op voorbereiden.



**Figuur 1.2:** Verdeling van de bandbreedte volgens applicaties/services [12, 4].

In realiteit weet een operator dit niet en zal hij met een beperkt budget keuzes moeten maken.

In deze upgradeplanning zal er rekening gehouden moeten worden met de verwachte toekomstige bandbreedtevereisten alsook met de impact van het falen van een netwerkcomponent. Eveneens zal men trachten om de interne verbindingen van het netwerk zo goed mogelijk te balanceren, zodanig dat de kans dat congestie optreedt kleiner is.

Het netwerk van Proximus dat in deze masterproef bestudeerd wordt is een meerlagige netwerk. Daarom zal er eerst in dit hoofdstuk wat meer informatie over meerlagige netwerken gegeven worden in sectie 1.1. Zoals hierboven reeds aangehaald is het falen van een netwerkcomponent een probleem, dit wordt verder besproken in sectie 1.2. In deze masterproef worden verschillende analyses gemaakt, één daarvan zijn de trafiekmatrices, in sectie 1.3 staat een inleiding over wat trafiekmatrices juist zijn. Ten slotte wordt in sectie 1.4 de verschillende pijlers van het onderzoeksproject tussen Proximus en UGent besproken. Hier wordt een verder verloop weergegeven van deze masterproef.

## 1.1 Meerlagige netwerken

Naast éénlagige netwerken, waarbij er één technologie in staat voor alle communicatie zijn er ook meerlagige netwerken [3]. Dit wil zeggen dat er verschillende lagen zijn die elk hun eigen functie hebben. Ze baseren zich hiervoor op het TCP/IP protocol stack. Elke laag kan bestaan uit verschillende technologieën. We identificeren twee belangrijke lagen: de

fysieke laag en de logische laag. In deze masterproef wordt er vooral ingegaan op wat er op de logische laag gebeurt.

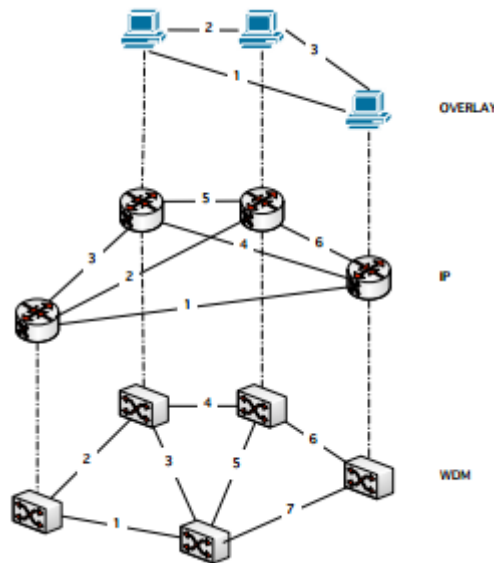
In de fysieke laag maken we onderscheid tussen de effectieve kabels en de hardware/software die gebruikt wordt om deze aan te sturen. Het laagste niveau maakt dus gebruik van optische vezelkabel, koperkabel, draadloze transmissies, . . . . Het niveau hierboven beheert deze communicatiekanalen. Hier worden technologieën zoals SONET/SDH, WDM, OTN gebruikt. Voor meer informatie over meerlagige netwerk architecturen: [3, 7].

Deze technologieën hebben allemaal hun voor- en nadelen en het is dus aan de operatoren om te kiezen hoe ze hun netwerk opbouwen. Het is niet ongewoon dat ook oudere technologieën voorkomen in deze architectuur. Het is namelijk niet altijd even realistisch (zowel praktisch als budgettair) om volledig over te schakelen naar de nieuwste beschikbare technologie.

Op de logische laag zien we vooral een explosie van IP-verkeer 1. Het computernetwerk bovenop een andere netwerk noemen we het ‘Overlay network’. Zo kan je stellen dat het internet origineel een overlay netwerk was op het bestaande telefoonnetwerk. Waar er vroeger een apart telefoonnetwerk (Plain old telephone service - POTS) was, kunnen we vandaag gebruik maken van Voice over IP (VoIP). Maar er zijn nog tal van andere diensten die over IP aangeboden worden zoals videogespreken, video conferencing, TV broadcast, Video on Demand (VoD). Figuur 1.3 geeft een voorbeeld van een meerlagig netwerk.

Het onderscheid tussen een overlay netwerk en een dienst is niet altijd even makkelijk te maken. Een VoD-dienst kan meerdere content- en cachingservers hebben. Het logisch geheel hiervan kan men als aparte laag beschouwen. De verschillende lagen maken het dus eenvoudiger om de architectuur van een netwerk te bestuderen en zorgen ook voor abstractie van de verschillende problemen binnen het netwerk.

Het upgraden van een meerlagige netwerk kan op verschillende lagen gebeuren. Bij het voorbeeld van een VoD-dienst kan het nuttig zijn om extra cachingservers te installeren om bepaalde links in het netwerk minder te belasten. Voor de upgradeplanning moet men dus rekening houden met de verschillende diensten en de vereisten van deze diensten.



**Figuur 1.3:** Een voorbeeld van een meerlagige netwerk [13].

## 1.2 Falen van netwerken

Om er voor te zorgen dat een netwerk zoveel mogelijk beschikbaar is, moet een netwerk dynamisch kunnen omgaan met het falen van een component er van. Het falen van een router, switch, stroom van een gebouw vol netwerkapparatuur, doorknippen van een kabel zijn veelvoorkomende problemen binnen netwerken [9, 15]. Het kan niet de bedoeling zijn dat een grote groep mensen geen toegang meer heeft tot het netwerk van een operator. Los van de financiële gevolgen voor het niet naleven van SLA's zijn er nog andere potentiële problemen: Wat als mensen noodoproepen over VoIP moeten maken? Wat als webwinkels niet meer beschikbaar zijn? Natuurlijk moeten operatoren nog altijd winst maken en zal er een afweging gemaakt moeten worden tussen de kost om availability te garanderen en de kost van eventuele downtime.

De oorzaken van falen zijn enorm divers en kunnen in volgende groepen onderverdeeld worden [15]:

- Gepland falen

Soms kan het nodig zijn om de software van bepaalde netwerkapparatuur te veranderen of bepaalde netwerkapparatuur te vervangen. Aangezien we op voorhand weten over welke nodes het gaat, kunnen we preventief actie ondernemen.



- Ongepland falen

De netwerk operator weet op voorhand niet wanneer of waar iets zal falen. Er zullen dus verschillende methoden voorzien moeten worden om er voor te zorgen dat het netwerk terug in een geldig toestand terechtkomt.

Dezelfde oorzaken kunnen ook opgedeeld worden volgens:

- Interne oorzaak

Het falen van software of hardware met andere woorden alles waar de netwerk operator controle over heeft.

- Externe oorzaak

Het uitvallen van de elektriciteit, aardbevingen, overstroming, het beschadigen van een kabel bij graafwerken, software-matige aanval op het netwerk, . . .

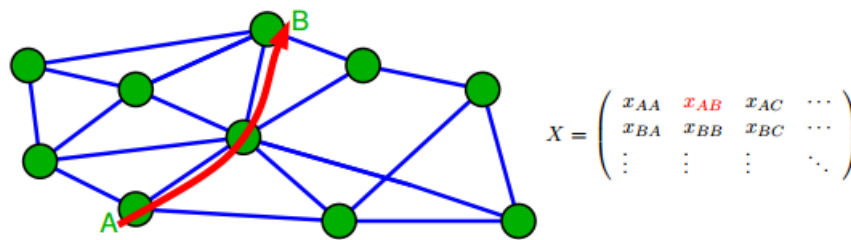
In [1] worden de oorzaken opgesplitst in kwaadwillig en niet-kwaadwillig. Zo zijn het gebruiken van een EMP-wapen en het storen van de draadloze signalen kwaadwillige oorzaken van het falen binnen een netwerk.

## 1.3 Trafiekmatrices

Operatoren moeten er voor zorgen dat in de toekomst hun netwerk beter gedimensioneerd is. Ze zullen niet alleen meer trafiek te verwerken krijgen maar ook op een kost-efficiënte manier moeten omgaan met hun netwerkupgradeplanning. Het is namelijk niet kost-efficiënt om op een bepaalde laag van het netwerk altijd maar meer capaciteit toe te voegen [7]. Ook de over-provisioning van capaciteit is een dure zaak die gedeeltelijk kan vermeden worden [2].

De upgradeplanning zal moeten rekening houden met een groot deel van de concepten die in de vorige punten besproken zijn. Specifiek zal resilience in meerlagige netwerken een belangrijke rol gaan spelen. In deze planning zal men ook rekening moeten houden met de kosten van het upgraden van een node. Is het nuttig om direct een grote investering te doen die op langere termijn een beter pay-off heeft of overwint het korte termijn denken? Moet een bepaalde node wel geüpgrade worden? Misschien kost het minder om bijvoorbeeld de kosten te betalen die vast hangen aan de SLA's bij het falen?

De klanten van operatoren verwachten een zo goed mogelijke service voor zo weinig mogelijk geld. Om operatoren beter te helpen om een keuze te maken in hun upgrade planning zal er in deze masterproef allereerst gefocust worden op het generen van trafiekmatrices op basis van reële data. Deze trafiekmatrices mappen het verkeer van een bestaand netwerk op een tweedimensionale  $n \times n$  matrix met  $n$  het aantal nodes. Een trafiekmatrix zal de bandbreedtevereiste bijhouden voor elke ingress node naar een egress node in het netwerk, over een bepaald tijdsinterval. Figuur 1.4 geeft een voorbeeld van een trafiekmatrix.



**Figuur 1.4:** Een voorbeeld van een trafiekmatrix [14]

In [14] worden twee definities gegeven van trafiekmatrices namelijk origin-destination (OD) en ingress-egress (IE). Bij een OD-trafiekmatrix zal men werken met de echte bron en de echte bestemming. Op IP-niveau gebruikt men IPv4/IPv6 adressen als bron/bestemming. Op deze manier kan men heel gedetailleerde dataflows bekijken binnen een netwerk. Dit soort matrix opbouwen is in de praktijk niet zo handig aangezien er  $2^{32}$  IPv4 adressen zijn en er dus enorm veel rekenkracht nodig is om deze te manipuleren. De IE-trafiekmatrix zal data aggregeren op basis van de nodes van een operator.

Het opstellen van deze matrix zal in de eerste plaats operatoren in staat stellen om een overzicht te krijgen van de huidige bandbreedte vereisten tussen verschillende links binnen hun netwerk. Verder kunnen deze matrices gebruikt worden om simulaties te maken. Maar kan men op basis van deze matrices verschillende grafieken genereren die de evolutie van trafiek binnen het netwerk weergeven. Zo is het mogelijk om te kijken wanneer er dagelijks of wekelijks peak-throughput is. Aangezien meeste trafiek door mensen gegenereerd wordt, is het niet verwonderlijk dat deze dezelfde dag- en nachtcycli volgt [14]. Merk hier wel op dat het om residentieel internetverbruik gaat, in hoofdstuk 5 blijkt dat er een duidelijk verschil is tussen residentieel en professioneel internetverbruik.

Een andere nuttig gegeven is de distributie van trafiek binnen het netwerk. Is trafiek

normaal verdeeld of staat een select aantal nodes in voor het grootste deel van de trafiek? Op deze manier is het ook mogelijk om via deze trafiekmatrices eventuele bottlenecks of congestie te identificeren.

Falen kan men ook via deze matrices simuleren door de bandbreedtevereisten tussen twee nodes op nul te zetten. Op deze manier kan dergelijke matrix gebruikt worden als input voor bijvoorbeeld een netwerksimulatiebibliotheek. Op diezelfde manier kunnen bepaalde bandbreedtevereiste dus ook geherouteerd worden over andere nodes, tijdens de recovery van het falen. Dankzij manipulaties aan de trafiekmatrices kunnen dus ook de impact van verschillende recovery mechanismen bestudeerd worden. Het identificeren van probleem-nodes kan door opvolging van de trafiekmatrices gebeuren (Anomaly Detection). Zo kan het voorvallen dat dezelfde node altijd faalt, dan is dit duidelijk te zien in de trafiekmatrices.

Nog een nut van trafiekmatrices is het voorspellen van trafiek. Op basis van alle beschikbare trafiekmatrices over een bepaalde periode, proberen we een voorspelling te maken over de toekomstige trafiekmatrices. Dit kan bijvoorbeeld gedaan worden door een huidig matrix te vermenigvuldigen met een verwachtingspercentage van extra trafiek, deze matrix kan dan weer gebruikt worden om een netwerk te simuleren. Op deze manier kunnen verschillende scenario's uitgewerkt worden. Dit is vooral nuttig om operatoren beter te helpen bij het opstellen van hun upgradeplanning. Als operatoren weten wat voor vereisten er in de toekomst zullen zijn dan kunnen ze gemakkelijk beslissen welke nodes belangrijk zijn om up te graden.

## 1.4 Onderzoeksproject

Deze masterproef omvat een sterke samenwerking met Proximus. Samen met UGent wordt er gewerkt aan een gemeenschappelijk onderzoekproject. Dit project heeft drie pijlers namelijk what-if scenario's, forecasting en content delivery networks ook wel CDN. Het doel van het project is om een flexible simulatie omgeving op te zetten waarin al deze pijlers aan bod komen. In deze omgeving kunnen dan verschillende simulaties gemaakt en geëvalueerd worden. Het resultaat van deze simulaties kan dan gebruikt worden om de upgradeplanning te verbeteren.

In het eerste deel van dit hoofdstuk werd een algemeen beeld gegeven van het probleem-domein waarbinnen deze masterproef kadert. De volgende alinea's geven meer uitleg over de drie pijlers en hun verband met deze masterproef.

What-if scenario's zijn hypothetisch probleemstellingen die interessant zijn om te onderzoeken. Een aantal voorbeelden hiervan zijn : wat als een bepaalde node in het netwerk faalt? Wat als een bepaalde link meer trafiek te verwerken krijgt? Wat als er  $x\%$  meer klanten zijn? Dergelijke analyses zijn broodnodig om een goede upgradeplanning te maken.

Als een node bij falen er voor zorgt dat een deel van netwerk onbereikbaar wordt, dan is het een goed idee om deze node te upgraden. Uiteraard zal de reële situatie nog complexer zijn, zo zal er rekening gehouden moeten worden met de kost van de upgrade versus de kost van het falen. Dergelijke worst-case scenarios zijn dus zeer interessant om te behandelen in de upgradeplanning, zodanig dat deze niet meer kunnen voorkomen. In sectie 1.2 wordt kort ingegaan op het falen binnen een netwerk. De trafiekmatrices die gemaakt worden in hoofdstuk 4 kunnen gebruikt worden om falen te simuleren en zelfs te detecteren. Verder kunnen uit de algemene conclusies van hoofdstuk 5 verwachtingswaarden opgesteld worden, om de impact van bijvoorbeeld  $x\%$  meer klanten te onderzoeken.

Het einddoel van forecasting is dat Proximus kan voorspellen hoeveel trafiek verwerkt moet worden door elke node op gelijk welk moment. Voor de upgradeplanning is het van essentieel belang dat een goede inschatting van trafiek gemaakt kan worden, om een zo kostenefficiënt mogelijk upgrade-pad te voorzien voor een bepaalde node. Aangezien een upgrade-pad over meerdere jaren kan gaan, is het dus nodig om een voorspelling over meerdere jaren te maken. In hoofdstuk 3 worden verschillende uitvoer mogelijkheden voorzien, die het opbouwen van historische informatie vergemakkelijkt. Deze informatie kan dan gebruikt worden om voorspelling te maken. Een mogelijke uitvoertype is trafiekmatrices, die besproken worden in hoofdstuk 4. Opnieuw kunnen de verwachtingswaarden uit hoofdstuk 5 gebruikt worden om te helpen met forecasting.

Content delivery zorgt ervoor dat afbeeldingen, video's en andere multimedia op een vlotte manier tot bij de eindgebruikers geraken. In plaats van één server die alle mogelijke content heeft, worden er verschillende cacheservers geïntroduceerd die een kopie van de content krijgen. Op deze manier kunnen eindgebruikers hun content afhalen bij de cacheservers en wordt zo het verkeer in het netwerk gereduceerd. Het is dus een goed idee om populaire content op dergelijke cacheserver te plaatsen. Hoeveel cacheservers moeten er dan geïntroduceerd worden en wat is de impact op het netwerk? Dergelijke vragen worden beantwoord in hoofdstuk 4.

In hoofdstuk 2 wordt er meer verteld over de data die Proximus geleverd heeft om deze masterproef te kunnen realiseren. In hoofdstuk 3 wordt een overzicht gegeven van het

programma dat ontwikkeld is tijdens deze masterproef. Op basis van de output van dit programma worden verschillende analyses gemaakt, deze worden besproken in hoofdstuk 4 en 5. Ten slotte wordt een conclusie geformuleerd in hoofdstuk 6, hierbij worden ook enkele toekomstperspectieven gegeven.

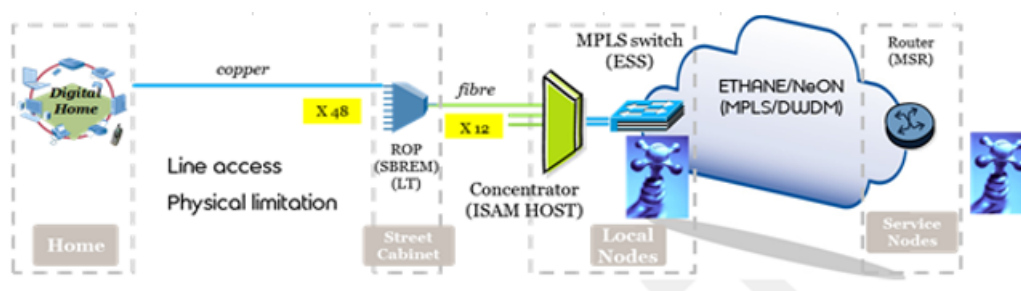
## Hoofdstuk 2

### Data

Het IP-netwerk van Proximus, genaamd Ethane, bestaat uit  $\approx 600$  nodes verspreid over België. Een node kan verschillende routers, switches en andere netwerkapparatuur bevatten. Het netwerk zelf wordt opgedeeld in drie hiërarchische lagen: *access*, *distribution* en *core*. Elke node heeft minstens één ethernet service switch ook wel ESS dat verbonden is met de regionale service nodes. Alle services nodes samen vormen het core netwerk van Proximus. De nodes die verbonden zijn met de services nodes, hebben één of meerdere concentrators/*aggregators*. Er zijn verschillende types van concentrators.

In figuur 2.1 is een overzicht te zien van deze netwerkhierarchie voor een ISAM-host. Dit is de meest voorkomende *aggregator* binnen het netwerk van Proximus. Deze *aggregator* aggregeert data van maximaal 12 *remote optical platforms* of ROP's. Deze ROP's bestaan uit maximaal 4 borden. Elk van deze borden heeft maximaal 48 klanten. Een ISAM-host kan maximaal 12 borden voorzien van toegang tot het netwerk, wat neerkomt op  $12 \times 48 = 576$  klanten. Wanneer er dus meer dan 576 klanten zijn, worden er meerdere ISAM-hosts geïnstalleerd in een bepaalde node. Er zijn dus verschillende mogelijke aggregatortypes, die instaan voor bepaalde diensten en elk een andere aantal klanten kan afhandelen. Een node kan op deze manier over verschillende *aggregator* types beschikken.

Voor elke node binnen het netwerk worden er continu metingen uitgevoerd. Hierbij wordt de gemiddelde throughput over vijf minuten voor zowel ingress als egress verkeer gemeten in bits per seconde. Deze metingen gebeuren voor alle *aggregators* maar ook voor alle diensten die gebruik maken van deze *aggregators*. Dit zijn *down-* en *upstream* metingen. Er wordt dus geen rekening gehouden met de throughput die nodig is om de data binnen het Proximus netwerk zelf te verplaatsen. Op deze manier kunnen de bandbreedtevereisten



**Figuur 2.1:** Proximus netwerkhiërarchie op basis van een ISAM-host.

per node uitgedrukt worden per *aggregator* type of per dienst. De enige uitzondering op deze regel zijn de services nodes.

De metingen zijn per dag verspreid over vier verschillende bestanden. In het volgende deel wordt een overzicht gegeven van deze bestanden.

## 2.1 Metingen per aggregator

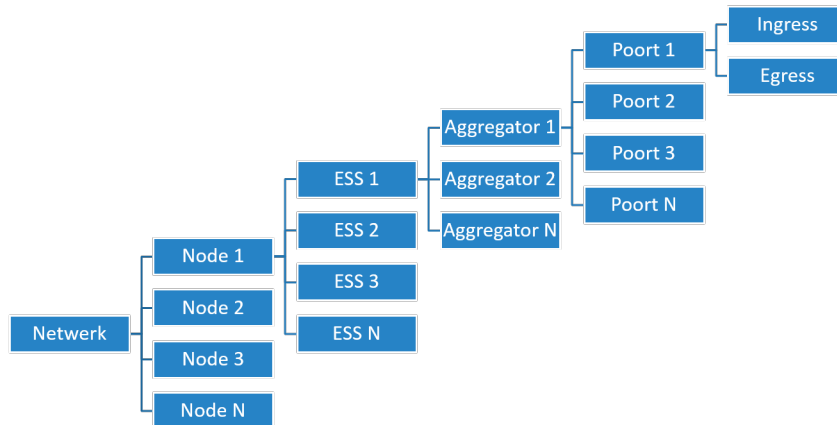
Eerst en vooral zijn er de metingen per *aggregator*. In figuur 2.2 wordt een hiërarchische overzicht gegeven van waar deze metingen zich juist bevinden in het netwerk. Op basis van deze figuur kunnen er verschillende niveaus gedefinieerd worden. Zo zullen alle metingen gebeuren op het poort-niveau. In deze tekst worden ook nog node-niveau en *equipment*-niveau gebruikt. Het *equipment*-niveau slaat dus op een bepaalde *aggregator* van een bepaalde node. Deze metingen kunnen verder opgesplitst worden in bedraad en draadloos.

### 2.1.1 Bedraad

Node	Port	Description	Logtime	Ingress b/s	Egress b/s
ena02als01	1/2/12	ap-ena02als01-h02als00009	31/03/17 00:00	3617232	180393764

**Tabel 2.1:** Een ingekort voorbeeld van een lijn uit de *aggregator* bestanden.

Elke node binnen het Ethane-netwerk wordt geïdentificeerd aan de hand van het zone-nummer en naam van de node. In de eerste kolom van tabel 2.1 staat er 02als01. Het



**Figuur 2.2:** Overzicht van metingen per *aggregator* binnen de Proximus netwerkhiërarchie

zonenummer 02 verwijst naar Brussel en 'als' naar de naam van de node. Het 01-suffix wijst er op dat dit informatie is over ESS 01.

Op basis van de beschrijving ('*description*' in de bestanden) wordt de informatie op deze lijn gekoppeld aan een *aggregator*. In dit geval is de *aggregator* 'h02als00009'. Elk aggregatortype heeft een identificatie aan de hand van een letter, een ISAM\_HOST krijgt 'h' als identificatie. Deze lijn is dus van toepassing op de negende ISAM\_HOST in de 02als-node en die is verbonden met ESS 01. De *logtime* geeft aan wanneer de meting gebeurt is. De metingen gebeuren normaal om de 5 minuten.

### 2.1.2 Klanten informatie

Er is ook informatie beschikbaar over het aantal klanten die gebruik maken van deze *aggregators*. Echter is voor bepaalde *aggregator* types informatie beschikbaar op *equipment*-niveau en voor andere informatie op node niveau. In tabel 2.2 wordt een overzicht hiervan gegeven.

### 2.1.3 Draadloos

Er zijn twee grote verschillen met de bedrade metingen. Voor draadloos worden andere aggregatortypes gebruikt. De identificatie van het *equipment*-niveau verloopt hier anders. Er zijn twee systemen die door elkaar gebruikt worden namelijk de codif-code of mobile cow-code. Op basis van deze codes kan onderscheid gemaakt worden tussen mobile backhaul voor Proximus en geleased lijnen aan klanten.



Poorttype	Identificatie	Klanten-niveau
AGW_HOST	a	Node
AGW_LEX	w	Node
FTTH_AGGR	f	Geen informatie
GENT_AGGR	c	Node
GELT	i	Node
GELT_BIS	g	Node
ISAM_HOST	h	Equipment
ISAM_LEX	l	Equipment

**Tabel 2.2:** Overzicht van alle bedrade *aggregator* types.

Node	Port	Description	Logtime	Ingress b/s	Egress b/s
ena02als01	2/02/2003	lp-ena02als01-lte03459434	31/03/17 00:00	32848	29094

**Tabel 2.3:** Een ingekort voorbeeld van een lijn uit de draadloze *aggregator* bestanden.

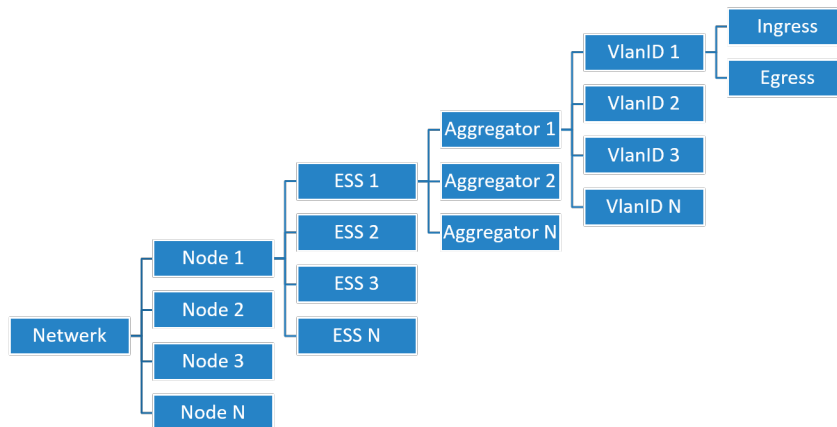
## 2.2 Metingen per service

Naast de metingen per aggregatortype zijn er ook metingen per VLAN in het netwerk van Proximus. In figuur 2.3 wordt opnieuw een hiërarchische overzicht gegeven van waar deze metingen zich juist bevinden in het netwerk. Door verschillende vlans te aggregeren, is het mogelijk om informatie te bekomen over de diensten binnen het netwerk. Er is dus een mapping nodig tussen de vlans en de diensten die aangeboden worden door Proximus. Enkele van de belangrijke diensten zijn 'fast internet access' (FIA) voor zowel residentieel als professioneel maar ook TV Broadcast<sup>1</sup> en TV Unicast.

### 2.2.1 Bedraad

Dit typebestand geeft een overzicht van de trafiekverdeling over de verschillende VLAN's die gebruikt worden in het Proximus netwerk. Voor een bepaalde node is de som van de ingress/egress waarden van alle diensten gelijk aan de som van de ingress/egress waarden van alle concentrators van die node. In tabel 2.4 staan er een voorbeeld voor het VLAN met id 3000. Buiten de verschillende poorttypes tussen bedraad en draadloos, is het enige verschil het bereik van de vlanID's. De nodeidentificatie, logtime, ingress en egress blijft

<sup>1</sup>Deze benaming heeft niets te maken met de manier waarop dit geïmplementeerd is.



**Figuur 2.3:** Overzicht van metingen per aggregator per service binnen de Proximus netwerkhiërarchie

hetzelfde.

Node	VlanID	Logtime	Ingress b/s	Egress b/s	Description Port
ena02als01	3000	31/03/17 00:00	18926544	695099	lag-ena02als01-h02als00001

**Tabel 2.4:** Een ingekort voorbeeld van een lijn uit de diensten bestanden.

## 2.2.2 Draadloos

In tabel 2.5 staat een voorbeeld uit de draadloze diensten bestanden.

Node	VlanID	Logtime	Ingress b/s	Egress b/s	Description Port
ena02als01	2012014	31/03/17 00:00	509548	3561494	lp-ena02als01-lte03552029-1

**Tabel 2.5:** Een ingekort voorbeeld van een lijn uit de draadloze diensten bestanden.

## 2.2.3 Klanten informatie

Ook voor de diensten is er informatie beschikbaar over klanten. In tabel 2.6 staat een overzicht van de beschikbare informatie. Dankzij deze informatie is het mogelijk om een gemiddelde contributie te maken per klant per node.

Het verschil tussen de twee type files is dat de *aggregator* metingen een algemeen beeld geven van de ingress/egress waarden binnen het netwerk. Dit kan per *aggregator* per node

Poorttype	Klanten-niveau	Diensten
FTTH_AGGR	Equipment	FIA_RES, TV, VoIP
GENT_AGGR	Node	FIA_RES
GELT	Node	TV
GELT_BIS	Node	TV
ISAM_HOST	Equipment	FIA_RES, FIA_PRO, TV
ISAM_LEX	Equipment	FIA_RES, FIA_PRO, TV

**Tabel 2.6:** Overzicht van de beschikbare klanteninformatie.

per vijf minuten. De metingen per service geven aan hoe deze ingress/egress waarden verdeeld zijn voor die *aggregator*. Hier kan men dus zien hoeveel trafiek er nodig is voor de verschillende diensten die aangeboden worden binnen het netwerk. Aangezien er over een *aggregator* data van meerdere diensten gestuurd kan worden, is het mogelijk om een overzicht per *aggregator* per dienst te maken.

Om een hele dag aan metingen te verwerken moeten er dus  $\approx 22.5$  miljoen lijnen ontleed worden. Hoe deze behandeld worden, wordt besproken in 3.

## 2.3 Geografische informatie

Er is ook informatie beschikbaar over de coppernetz van Proximus. De metingen per dienst kunnen hieraan gekoppeld worden om een beeld te krijgen van de geografisch evolutie doorheen de dag. Dit wordt uitgebreid besproken in 5.4.

## 2.4 Voorkomende problemen

Het is mogelijk dat een node geen metingen heeft voor een bepaald tijdsinterval. Hiervoor kunnen verschillende oorzaken zijn die niet bepaald kunnen worden op basis van de input-data. Zo kan het zijn dat de meetapparatuur uitvalt, stroomonderbreking bij een bepaalde node of andere oorzaken van falen binnen het netwerk, .... Deze metingen mogen niet beschouwt worden als 0-metingen want het is mogelijk dat er in een bepaald tijdsinterval geen of heel weinig trafiek is.

Sommige lijnen wijken af van de vooropgestelde waarden. Zo zijn er ook metingen van

controle signalen. Soms zijn er kleine afwijkingen zoals bijvoorbeeld ‘ltp’ in plaats van ‘lte’. Soms ontbreekt ook informatie zoals bij ‘lp—1’, tussen de dashes zou in feite het aggregatortype moeten staan. Dergelijke onvolledige lijnen zijn niet nuttig voor verdere analyse en worden dan ook genegeerd.

Een inputbestand heeft normaal metingen voor één dag. Voor een deel van de *aggregator* inputbestanden zijn er metingen van de volgende dag opgenomen. Het gaat hier wel telkens over hetzelfde tijdsframe.

Een ander probleem is het ontbreken van klantenaantallen voor bepaalde ISAM's. Zo zijn er 430 *equipments* waarvoor wel metingen zijn maar geen klantenaantallen. Veel van deze metingen zijn nul waarden of controle signalen ( $\leq 2$ kbps). Na overleg met Proximus blijkt dat het gaat om nieuw geïnstalleerde *equipments*, deze hebben nog geen klanten. In hetzelfde bestand zijn er ook 166 *equipments* waarvoor geen informatie beschikbaar is over de diensten.

## Hoofdstuk 3

### Tool

In deze masterproef worden, op basis van de data besproken in hoofdstuk 2, verschillende analyses gemaakt. Er zijn twee categorieën van analyses namelijk trafiekmatrices en trends. Deze worden respectievelijk in hoofdstuk 4 en 5 besproken. Deze analyses geven Proximus meer informatie voor het opstellen van hun upgradeplanning. Om deze analyses te kunnen maken is er dus een tool nodig die er voor zorgt dat de data op een makkelijke manipuleerbare manier voorgesteld kan worden.

Het verwerken van deze data is opgesplitst in drie stappen: het parsen, analyseren en opslaan. Dit gebeurt met een Maven-project in Java. Figuur 3.1 geeft een overzicht van de verschillende stappen in de *workflow*. In dit hoofdstuk wordt uitgebreid in gegaan op de eerste twee delen van deze *workflow*, het parsen en het eerste deel van de analyse-fase namelijk aggregaties maken.



**Figuur 3.1:** Workflow van applicatie voor het verwerken van data.

### 3.1 Parsen

Bij het parsen worden input-bestanden ingelezen en lijn-per-lijn verwerkt. Niet elke lijn is nuttig voor de analyse-fase. Zo zijn er bepaalde nodes die op vraag van Proximus

genegeerd moeten worden. Het gaat hier over nodes die instaan voor *disaster recovery* en datacenters. Deze zouden tijdens de analyse-fase een vertekend beeld kunnen geven van de trafiek binnen het netwerk van Proximus. De nodes die genegeerd mogen worden, staan allemaal samen in een ignore-configuratiebestand.

Er wordt dus eerst gekeken naar de naam van de node, als deze voorkomt in de ignore-bestand wordt deze lijn genegeerd. Vervolgens wordt de locatie en het aggregatortype verwerkt. Op basis van deze informatie kan men bepalen of diezelfde node reeds verwerkt geweest is of niet. Uiteindelijk wordt er een meting toegevoegd aan de nieuwe node of aan de reeds verwerkte node.

De algemene methodes zoals het parsen van het aggregatortype, datum, locatie en ignore-list checken worden aangeboden in de superklasse IParser. Uiteraard zijn er ook voor elk bestand verschillen. Zo bevat de dienstenbestanden meer kolommen dan de aggregatorbestanden. Ook staat niet alle data op dezelfde plaats. Om dit op te lossen is er voor elk type file een configuratiebestand waarin staat hoe dergelijke file verwerkt moet worden. Zo zijn er enkele variabelen die aangeven waar welke data staat. Er is dus per type inputbestand een concrete implementatie van IParser.

In 2.4 werden reeds enkele problemen met de data besproken. Naast het negeren van niet nuttig-lijnen zullen dus ook afwijkende lijnen genegeerd moeten worden. Voor de applicatie is het dus een preconditionie dat de inputdata een correct formaat heeft.

Het resultaat van het parsen van een bepaald bestand, is een lijst van *equipment*-objecten. Voor elk *equipment* is er ofwel een lijst van metingen ofwel een lijst van diensten die op hun beurt een lijst van metingen hebben.

### 3.1.1 Configuratiebestanden

Om het parsen in goede banen te leiden zijn er verschillende configuratiebestanden. Deze bestanden zorgen ervoor dat de parser configureerbaar is tot op een zeker punt. Aangezien het aantal kolommen verschilt voor aggregator en dienstenmetingen, moet het programma weten in welke kolom welke informatie staat. Per bestandstype is er een configuratiebestand dat dit aangeeft. Bijvoorbeeld zo geeft *c\_ingress = 9* aan dat de ingresswaarde te vinden is in de negende kolom, dit is zero-based.

In tabel 2.1 werd reeds een voorbeeld gegeven van een node-identificatie, namelijk ena02als01.

Uiteraard moet dit ook geparseerd worden en is het mogelijk dat dit formaat in de toekomst verandert. Om dit op te lossen is er een algemeen configuratiebestand met informatie over velden die voor alle inputbestanden dezelfde zijn. Om bijvoorbeeld de naam van de node te identificeren zijn volgende lijnen toegevoegd, `location_start=5` en `location_end=8`. Dit is de zero-based index van de node-identificatie. De parser zal vervolgens een substring nemen op basis van deze indexen. De dienstenbestanden hebben een extra veld tegenover de aggregatorbestanden, namelijk het `VlanID`. Het configuratiebestand heeft dan ook een extra lijn met de index van deze kolom.

De ignore-file heeft twee lijnen namelijk de `ignorenodes` en `servicenodes`. Zoals eerder vermeldt zijn er enkele nodes die op vraag van Proximus genegeerd moeten worden. Dergelijke nodes kunnen toegevoegd worden aan de `ignorenodes` variabele. Deze nodes kunnen uit de volledige node-identificatie bestaan of een deel ervan. Zo zorgt het toevoegen van `02` voor het negeren van alle nodes die `phonezone` gelijk aan `02` hebben. In alle `servicenodes`<sup>1</sup> worden metingen van drie ESS'en genegeerd. Deze switches behandelen geen verkeer van zogenaamde lokale klanten<sup>2</sup> en zouden dus een vertekend beeld kunnen geven van de trafiekstromen binnen het netwerk.

Zoals eerder aangegeven in sectie 2.2 is er een mapping nodig tussen de verschillende vlans en de dienst waarvoor ze trafiek versturen. Hier voor is er een configuratie bestand met als sleutel het `vlanID` en als waarde een `serviceID`. De `serviceID`'s zijn oplopend toegewezen. Wanneer er bijvoorbeeld staat `10 = 0` betekent dit dat `vlan` met id `10` toegewezen wordt aan een service met id `0`. Om de output van het programma duidelijker te maken is er ook een configuratiebestand waarbij aan elke `serviceid` een naam kan toegewezen worden, een voorbeeld hiervan is `0 = FIA_RES`.

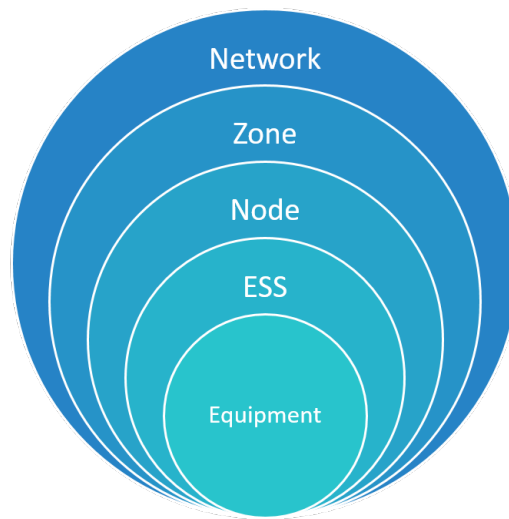
## 3.2 Aggregaties

Met de output van de parser is het mogelijk om verschillende aggregaties te maken. Deze aggregaties zijn op te delen over de verschillende hiërarchische niveaus binnen het netwerk. In figuur 3.2 wordt een overzicht gegeven van de verschillende aggregatie niveaus. Merk hier op dat voor het netwerk en node niveau ook een aggregatie per aggregator type kan gemaakt worden.

---

<sup>1</sup>In hoofdstuk 2 werd het concept van `servicenodes` reeds besproken.

<sup>2</sup>Een lokale klant is rechtstreeks aangesloten op een bepaalde node.



**Figuur 3.2:** Een overzicht van mogelijke aggregaties.

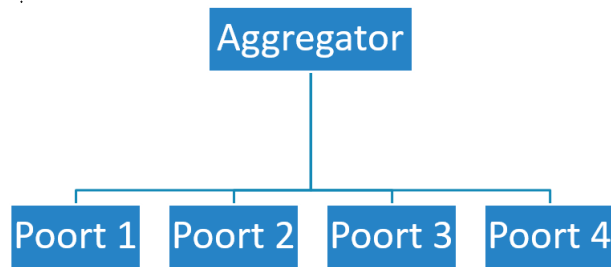
Een aggregatie op zone-niveau betekent eigenlijk een aggregatie van een lijst van nodes. Dit wordt specifiek gebruikt in het kader van het SIMBA-project, hierbij wordt de backbone van Proximus verdeeld in vijf zones. Om te weten welke node bij welke zone hoort, is er een configuratiebestand geïntroduceerd die de mapping maakt tussen de phonezone en simbazone. Het is echter mogelijk om in deze mapping extra zones te definiëren zonder rekening te houden met de simbazones.

### 3.3 Aggregatie problemen

Zoals eerder aangegeven kan het voorvallen dat er voor bepaalde tijdstippen meerdere of geen meetwaarden zijn. Tijdens het parsen zelf wordt hier geen rekening mee gehouden. Het is pas in de analyse-fase dat er een aggregatie van de data gemaakt worden. Allereerst wordt het probleem van meerdere meetwaarden op hetzelfde tijdstip opgelost door de som te nemen van al deze meetwaarden voor die aggregator, zie figuur 3.3.

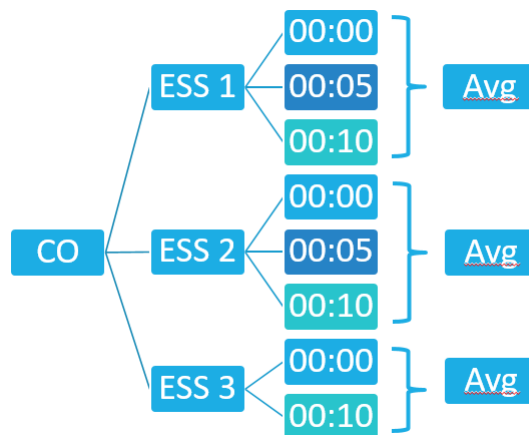
Een andere probleem is wanneer er een aggregatie voor elke x minuten nodig is. Het kan zijn dat er voor een bepaald tijdstip geen meetwaarde is. Er wordt dan een gemiddelde genomen over de beschikbare tijdstippen voor dat interval. In figuur 3.4 staat hiervan een voorbeeld. Hierdoor kan de uiteindelijke throughput waarde afwijken van wat men zou verwachten. Dit zal zeker het geval zijn wanneer er bijvoorbeeld slechts één meetwaarde beschikbaar is dat op zich al een uitschieter is. Dit zou kunnen opgelost worden door





**Figuur 3.3:** Metingen over verschillende poorten voor hetzelfde tijdstip.

bijvoorbeeld het gemiddelde te nemen over verschillende dagen voor dat tijdsinterval om zoveel mogelijk de impact van die uitschieter te minimaliseren. Een andere optie is om uitschieters volledig te negeren.



**Figuur 3.4:** Aggregatie maken over een groter tijdsinterval dan de beschikbare vijf minuten.

### 3.3.1 Klanten

Naast het parsen van metingen van aggregators en diensten is het ook mogelijk om informatie over klanten hierbij te betrekken. Op deze manier is het mogelijk om voor de verschillende aggregaties niveaus, een contributie per klant te bekomen. Echter is er niet voor alle aggregator types alle klanteninformatie beschikbaar. Dit werd reeds besproken in subsectie 2.1.2 en 2.2.3.

Het parsen van klanteninformatie gebeurt voor het maken van de aggregaties en na het parsen van de lijst van *equipment*. Bij het parsen zal de input van de klantenaantallen worden

gemanipuleerd zodanig dat er voor alle aggregatortypes, klanteninformatie beschikbaar is op *equipment*-niveau. Een voorbeeld hiervan zijn de AGW\_LEX *equipments*. Bij node 02als zijn er 835 AGW\_LEX klanten, in tabel 3.1 staat een overzicht van deze manipulatie. Merk op dat het gemiddelde wordt genomen en er dus in dit geval één klant verloren gaat.

Node	Niveau	Aantal klanten
02ALS0 - AGW_LEX	Node	835
↓	↓	↓
02als01 - AGW_LEX - 1	Equipment	278
02als01 - AGW_LEX - 2	Equipment	278
02als01 - AGW_LEX - 3	Equipment	278

**Tabel 3.1:** Een voorbeeld van de inputmanipulatie bij aggregator klanten.

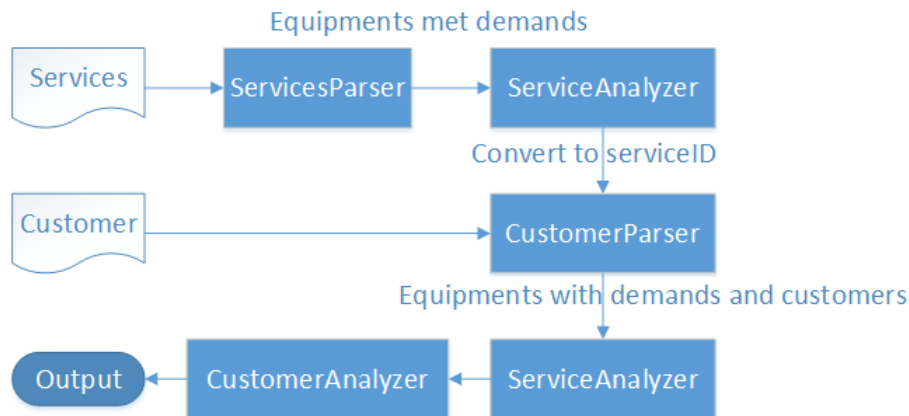
Wanneer er dan een bepaalde aggregatie gemaakt wordt, zal deze aggregatie rekening moeten houden met de klantenaantallen. Dit gebeurt door de som te nemen van alle klanten die onder een hiërarchische niveau vallen, uiteraard kan dit enkel per aggregatortype. Zo zal voor het netwerk niveau de som genomen van alle klantenaantallen in alle *equipments*. Hetzelfde gebeurt bij de diensten. Na het maken van een bepaalde aggregatie, zijn er dus enkel klantenaantallen voor dat niveau. Om dan ook nog eens een contributie per klant toe te voegen, worden de metingen van dat niveau gedeeld door het aantal klanten.

### 3.3.2 Voorbeeld van een mogelijke applicatieflow

In deze subsectie wordt een voorbeeld gegeven van een mogelijke applicatieflow. Het doel is om een aggregatie te maken per central office (ookwel node) per dienst en per klant voor ISAM\_HOSTS.

Het eerste wat gebeurt in deze flow is het parsen van een dienstenbestand. Er wordt dus een instantie aangemaakt van de concrete klasse ServicesParser. Bij het maken van een parser moet er een configuratie-bestand opgegeven worden, dit zijn de specifieke configuratiebestanden per type input-bestand. Verder zal in de superklasse het algemeen en ignore-configuratiebestand ook ingelezen worden. In subsectie 3.1.1 werden alle configuratiebestanden reeds besproken.

Dan wordt het input-bestand lijn per lijn verwerkt, zie sectie 3.1. Het resultaat van het parsen is een lijst van equipments waarbij elke equipment een map heeft met als sleutel het



**Figuur 3.5:** Workflow van applicatie voor het maken van een per CO per dienst per klant overzicht.

vlanID en als waarde het *service*-object. Dit *service*-object heeft een lijst van metingen, maar ook bijvoorbeeld de naam van de service.

Bij de volgende stap worden de vlanID's omgezet naar serviceID's. Bij deze conversie wordt dus de som genomen van alle metingen voor de vlans die tot een bepaalde service horen per tijdsinterval. Op deze manier is het eindresultaat een lijst van equipments met een *demands*-map die als sleutel het serviceID heeft en als waarde *service*-objecten heeft. Deze *service*-objecten bevatten de metingen.

Vooraleer een aggregatie over deze nodes kan gemaakt worden moet de klantenaantallen toegevoegd worden. Zodanig dat deze ook verwerkt kunnen worden tijdens het aggregeren van de data. Hiervoor wordt de CustomerParser gebruikt. De klantenaantallen worden op *equipment*-niveau toegevoegd per dienst.

Vervolgens wordt de aggregatie gemaakt op basis van het ingestelde aggregatieniveau. Om dit per node te kunnen doen wordt eerst een map aangemaakt. Dit gebeurt op basis van Java 8 streams. Het resultaat van deze mapping heeft als sleutel de node-identificatie zonder ESS (bijvoorbeeld 02als) en als waarde een lijst van *equipments* voor deze node (02als heeft verschillende ISAM\_HOSTS). Op basis van deze lijsten worden de aggregaties gemaakt. Indien de equipments over klantenaantallen beschikken dan worden deze ook geaggregeerd. Het resultaat van deze operaties zijn dus enerzijds een lijst equipments met demands per dienst per node. De *service*-objecten hebben een referentie naar een Customer-object dat de totale klantenaantallen heeft voor die dienst.

Als laatste stap worden de contributies per klant toegevoegd. Dit gebeurt op basis van de geaggregeerde metingen, die in de vorige stap gemaakt werden. In figuur 3.5 wordt een overzicht gegeven van deze opeenvolgende stappen. Eerst worden de equipments met hun demands ingelezen op basis van de inputbestand. Vervolgens worden de vlanID's omgezet naar serviceID's. Dan wordt er klanteninformatie toegevoegd op equipmentniveau. Ten slotte wordt de aggregatie op het ingestelde aggregatieniveau gemaakt.

## 3.4 Gebruik

De tool zal in de toekomst door Proximus gebruikt worden om de verschillende aggregaties en andere analyses te maken. De tool is in Java 8 ontwikkeld en maakt gebruik van Java Streams. Met andere woorden gelijk welke Java 8 omgeving met JDK kan gebruikt worden om de output te genereren.

Verschillende main-klassen werden aangemaakt zodanig dat enkel de parameters moeten aangepast worden om het programma uit te voeren. Een lijst van de bestandsnamen van de inputbestanden zal nodig zijn om de overzichten te genereren. Deze parameters zijn niet opgeven in een configuratie-bestand, er zal dus compilatie van de main-klassen nodig zijn.

In tabel 3.2 is de gemiddelde uitvoeringstijd terug te vinden voor enkel aggregaties. Deze metingen zijn uitgevoerd op een Windows 10 machine met een i7-920 processor. Merk op dat het RAM verbruik van het programma fluctueert tussen de  $\approx 3 - 4$  GB.

Soort inputbestand	Aggregatielevel	Gem. tijd (seconde)
Aggregator Bedraad	Per node per aggregatortype	51
Aggregator Draadloos	Per node per aggregatortype	35
Diensten Bedraad	Per aggregatortype voor network	54
Diensten Bedraad met klanten	Per aggregatortype voor network	60

**Tabel 3.2:** De gemiddelde uitvoeringstijd voor enkele aggregaties per inputbestand

## 3.5 Uitbreidbaarheid

In de tool is een zekere flexibiliteit ingebouwd. Zo is het mogelijk dat de node-identificatie van formaat veranderd. In de volgende lijst worden enkele van deze mogelijkheden besproken.

- toevoegen van een aggregatortype

Er zijn twee plaatsen waar het aggregatortype actief gebruikt wordt, namelijk in de *parsers* en bij de enum *aggregatortype*. Om een aggregatortype toe te voegen moet er dus eerst een extra enum-waarde toegevoegd worden. Vervolgens moet er in de abstracte klasse van de parsers het juiste aggregatortype ingesteld worden, dit komt neer op het toevoegen van een nieuwe letter. In tabel 2.2 werd reeds een voorbeeld hiervan gegeven.

- toevoegen van een nieuwe dienst

Voor het mappen van VLAN's op diensten is er een configuratiebestand beschikbaar. Op basis van dit bestand gebeurt de mapping. Als in de toekomst bepaalde VLAN's aan andere diensten toegewezen worden dan hoeft men enkel het configuratie-bestand aan te passen. Indien men een nieuwe dienst wil introduceren met andere VLAN's kan dit ook via dit configuratie-bestand.

- aanpassen van de node-identificatie

Zolang de node-identificatie blijft bestaan uit een *phonezone*, naam en ESS-nummer is er geen enkel probleem. Via het algemene configuratiebestand, besproken in subsectie 3.1.1, kan via indexen aangegeven worden waar wat staat. De eigenschappen van deze waarden moeten wel dezelfde blijven. Het ESS-nummer is intern opgeslagen als een integer. Als het ESS-nummer met letters zou uitgebreid worden dan zal er een exceptie opgeworpen worden, Java is immers een sterk getypeerde taal. In dit soort gevallen zullen de achterliggende domein-objecten aangepast moeten worden.

## Hoofdstuk 4

# Opbouwen van Trafiekmatrices

In deze masterproef worden er trafiekmatrices gemaakt op basis van trafiekstromen binnen het netwerk van Proximus. Het concept van trafiekmatrices werd reeds besproken in hoofdstuk 1. Deze trafiekmatrices worden dan als input gebruikt voor een netwerksimulatiebibliotheek. Deze trafiekmatrices dienen als basis om de huidige netwerkinfrastructuur beter te benutten en een betere upgradeplanning op te stellen.

Op basis van de geanalyseerde data, besproken in hoofdstuk 2, is het dan mogelijk om met de huidige bandbreedtevereisten trafiekmatrices op te stellen. Dit kan opnieuw op de verschillende aggregatieniveaus. Om een trafiekmatrix op te stellen is er een bron, bestemming, tijdstip en *throughput* nodig. Per node zijn de bandbreedtevereisten per dienst beschikbaar. De bron van de data is onbekend, aangezien we enkel beschikken over de lokale vereisten. Om dit op te lossen worden er *injection points* geïntroduceerd. Deze *injection points* zullen dienen als bron van de bandbreedtevereisten. Hierna worden de demands verspreid over deze *injection points*.

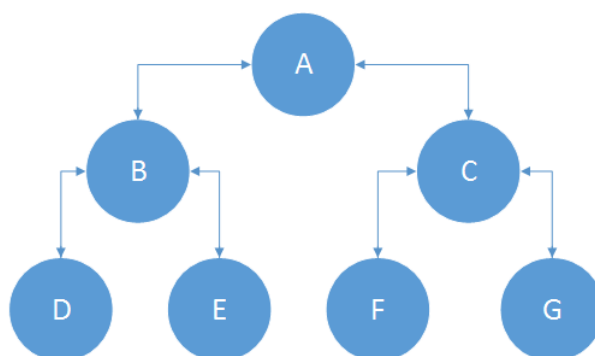
In eerste instantie zal er dus een  $1 \times 600$  matrix gemaakt worden. Waarbij er dus e n *injection point* is voor 600 nodes. Wanneer er een matrix met meerdere *injection points* gemaakt wordt, is het mogelijk om verschillende permutaties van deze trafiekmatrices te maken. Een *injection point* is een node die als bron zal dienen van de trafiek binnen het netwerk. Het verleggen van een *injection point* in een netwerk kan een grote impact geven. Zo is het mogelijk dat bepaalde links minder belast worden en dat de impact van falen in het netwerk kleiner wordt door beter gekozen *injection points*. Het aanleveren van *injection points* die de impact op zowel congestie en falen minimaliseren kan in de upgradeplanning gebruikt worden om het netwerk beter te dimensioneren.

Het is dan ook zinvol om te bestuderen wat er met het een netwerk gebeurt wanneer we bepaalde *injection points* gaan verleggen. Op basis van de vereisten per dienst worden er trafiekmatrices gegenereerd. De impact van de configuratie wordt dan gecontroleerd aan de hand van een netwerksimulatiebibliotheek. Deze simulatiebibliotheek zal in staan voor de *routes* te berekenen waarover data verstuurd wordt. Maar zal ook de link-belasting zelf berekenen per link in het netwerk.

In dit hoofdstuk worden eerst enkel voorbeelden van trafiekmatrices gegeven op basis van een voorbeeldnetwerk in figuur 4.1. Vervolgens worden alle mogelijke simulaties gegeven voor dit netwerk. Tenslotte wordt een verband gemaakt tussen dit concept en de upgraderplanning.

## 4.1 Theoretisch Voorbeeld

Op basis van figuur 4.1 worden enkele voorbeelden van trafiekmatrices uitgewerkt. Dit is een eenvoudig netwerk dat bestaat uit zeven nodes. Allereerst beginnen we met één *injection point*, dit wil zeggen dat alle trafiek van het netwerk afkomstig is van dit punt. Uiteraard zijn er ook fictieve vereisten of *demands* nodig van een bepaalde dienst X. In tabel 4.1 vinden we deze vereisten, zo heeft bijvoorbeeld Node E een bandbreedtevereiste van 15 eenheden voor dienst X. In de realiteit zal elke node ook de services nodes een zekere bandbreedtevereiste hebben voor de lokale klanten.



**Figuur 4.1:** Een netwerk dat als basis dient voor een aantal voorbeeld simulaties.

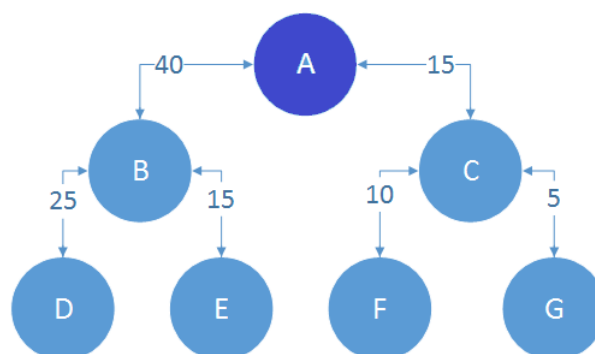
Indien men de bandbreedtevereisten van de verschillende nodes voor een bepaalde dienst kent is het opstellen van een trafiekmatrix met één *injection point* dus relatief eenvoudig. In dit geval zal Node A instaan voor het al het verkeer van dienst X. Node A zal dus

Dienst X	Demands
Node D	25
Node E	15
Node F	10
Node G	5

**Tabel 4.1:** Egress Demands

minstens een capaciteit nodig hebben van  $40 + 15 = 55$  eenheden. Figuur 4.2 geeft het volledig overzicht na toepassen van deze trafiekmatrix in tabel 4.2.

Dienst X	Node D	Node E	Node F	Node G
Injection Point A	25	15	10	5

**Tabel 4.2:** Trafiekmatrix met A als *injection point***Figuur 4.2:** Voorbeeld simulatie met trafiekmatrix 4.2

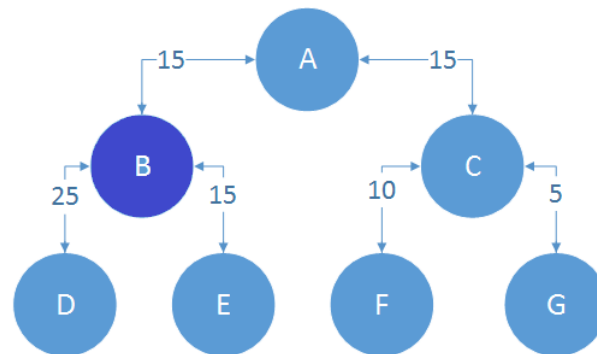
Wat gebeurt er nu als er een ander *injection point* dan A gekozen wordt? Bijvoorbeeld Node B? De trafiekmatrix zal er nagenoeg hetzelfde uitzien bij één *injection point*. Het netwerk zelf zal al een ‘grote’ verandering ondergaan. Node A zal nu slechts een totale capaciteit nodig hebben van  $15 + 15 = 30$  zie figuur 4.3. Indien nu Node A zou falen dan is er geen communicatie meer mogelijk met Node C, F en G aangezien al het verkeer voor deze nodes over Node A loopt.

Uiteraard willen we niet enkel de situatie weten met één *injection point*, maar ook die met meerdere *injection points*. In een eerste fase worden de *injection points* afwisselend toegewezen. Wanneer er twee nodes aangewezen zijn als *injection points* voor dienst X



Dienst X	Node D	Node E	Node F	Node G1
Injection Point B	25	15	10	5

**Tabel 4.3:** Trafiekmatrix met B als *injection point*

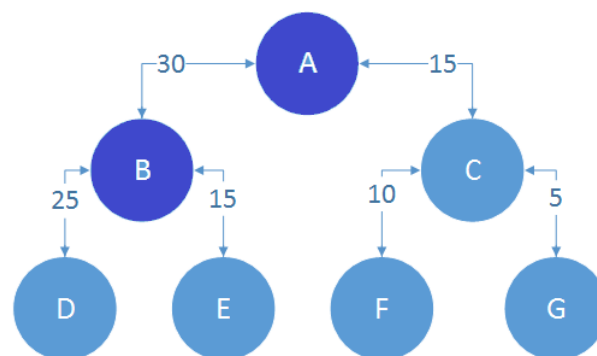


**Figuur 4.3:** Simulatie met Node B als *injection point* voor alle demands zie tabel 4.3.

krijgen we volgende trafiek matrix 4.4. Dit dient dan opnieuw als invoer voor simulaties. Node A heeft nu een totale capaciteit nodig van 45 eenheden.

Dienst X	Node D	Node E	Node F	Node G
Injection Point A	25	0	10	0
Injection Point B	0	15	0	5

**Tabel 4.4:** Trafiekmatrix met twee *injection points* afwisselend toegewezen



**Figuur 4.4:** Simulatie op basis van tabel 4.4

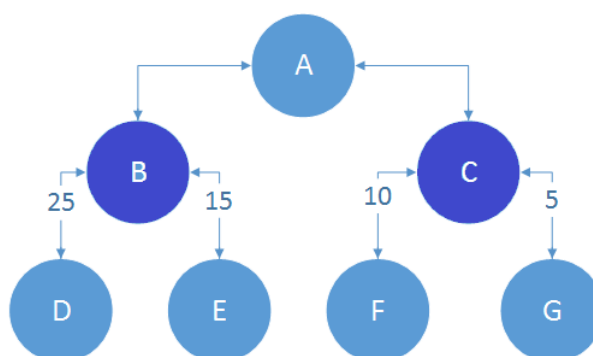
Het is hier dus ook mogelijk om een trafiekmatrix op te stellen waarbij Node A niet belast wordt. Dit kan door Node B en C op te laten treden als *injection points*. Hierdoor wordt

figuur 4.5 verkregen. Dezelfde figuur als 4.4 zonder capaciteit voor A. Zoals eerder vermeld is dit scenario in de realiteit onwaarschijnlijk.

Mocht Node A falen in dit scenario dan zal de communicatie tussen alle nodes nog altijd perfect verlopen. Aangezien Node A niet gebruikt wordt. In het vorige scenario, namelijk figuur 4.4, is er wel een probleem. Node A staat namelijk in voor Node D en Node F maar ook voor al het verkeer van Node B naar Node G. In dat geval zorgt het falen van Node A ervoor dat Node C, F en G niet meer bereikbaar zijn. De demands van D, F en G zullen niet voldaan zijn.

Dienst X	Node D	Node E	Node F	Node G
Injection Point B	25	15	0	0
Injection Point C	0	0	10	5

**Tabel 4.5:** Trafiekmatrix met twee *injection points* zonder belasting voor Node A.



**Figuur 4.5:** Simulatie op basis van tabel 4.5

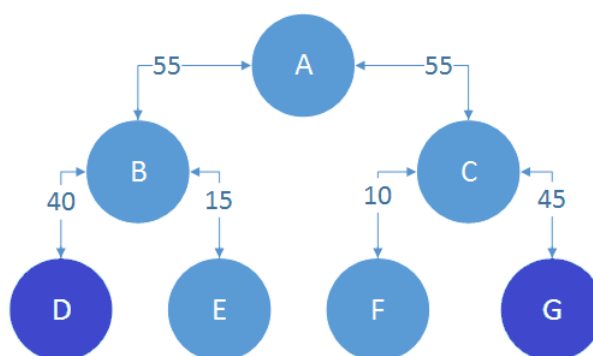
In een volgende fase is het de bedoeling om op zoek te gaan naar alle mogelijke nuttige permutaties van deze trafiekmatrix. Vervolgens kunnen we dan weer een *injection point* toevoegen en op zoek gaan naar alle permutaties van de trafiekmatrix met extra *injection point*. Voor elke node in de trafiekmatrix zijn er minstens het aantal *injection points* aan keuzes. De volgorde is belangrijk want de trafiekmatrix van 0001<sup>1</sup> en 1000 zal een andere impact hebben. Dit is dus een herhalingvarantie, want de volgorde is belangrijk en meerdere nodes kunnen over dezelfde *injection point* gaan. Voor deze  $2 \times 4$  matrix zijn er dus  $2^4 = 16$  mogelijkheden.

<sup>1</sup>0 en 1 als mogelijke keuzes. Nodes 1,2,3 krijgen 0 toegewezen en node 4, 1.

Het koppelen van naar-nodes aan van-nodes die enorm ver uit elkaar liggen in de topologie is op zich niet nuttig. Node G die als *injection point* moet instaan voor Node D en E zal enkel voor meer belasting zorgen van de links tussen Node G en Node B. In figuur 4.6 heeft Node A bijvoorbeeld een totale capaciteit nodig van  $55 + 55 = 110$  eenheden.

Dienst X	Node D	Node E	Node F	Node G
Injection Point G	25	15	0	0
Injection Point D	0	0	10	5

**Tabel 4.6:** Trafiekmatrix waarvan de *injection points* een grote hop-afstand hebben tegenover de vereisten van dienst X.



**Figuur 4.6:** Simulatie op basis van tabel 4.6

Het is dus geen slecht idee om op basis van de hop-afstand bepaalde *injection points* te elimineren om het aantal mogelijke trafiekmatrices te verkleinen. Het simuleren van 10 *injection points* over 600 nodes geeft dus  $10^{600}$  mogelijkheden, het is dus zeer nuttig om bepaalde mogelijkheden uit te sluiten. Een andere optie is om bijvoorbeeld met een vooraf gedefinieerde lijst van *injections points* te werken en de vereisten per node evenwichtig te verdelen over deze *injection points*.

## 4.2 Simulaties

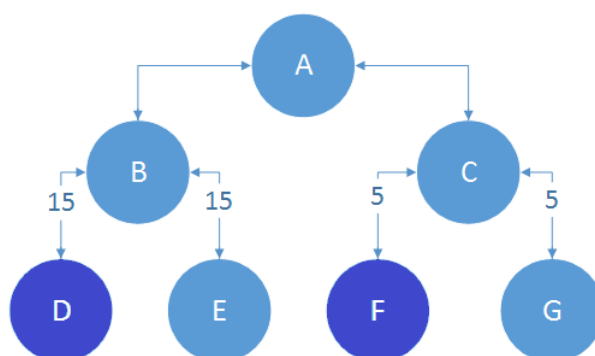
Het uiteindelijk doel is om de *injection points* zodanig te plaatsen dat de huidige netwerk-infrastructuur beter benut wordt en de impact van falen geminimaliseerd wordt. Hiervoor zullen dus verschillende trafiekmatrices moeten gesimuleerd worden.

Voor het netwerk in figuur 4.1 staan alle mogelijke simulaties met twee *injection points* in bijlage A. Hierbij zijn de *injection points* nog niet toegewezen aan reële nodes. Elke unieke combinatie van twee nodes kan gebruikt worden als *injection points*,  $\binom{7}{2} = 21$ . Voor dit netwerk volledig te simuleren met demands voor vier nodes zijn er dus  $21 * 16 = 336$  simulaties nodig.

Op basis van deze simulaties is het mogelijk om de combinatie te vinden waarvoor de totale link belasting zo laag mogelijk is. Dit komt dus neer op het iteratief overlopen van alle mogelijke permutaties. Het is logisch dat hoe dichter de demands bij hun bron voldaan worden, hoe minder netwerkverkeer er nodig is. Wanneer Node D en F zelf instaan voor hun eigen demands moet er enkel nog 20 eenheden verstuurd worden over het netwerk. De belasting op alle links van het netwerk is dan 40, dit is duidelijk te zien in figuur 4.7.

Dienst X	Node D	Node E	Node F	Node G
Injection Point D	25	15	0	0
Injection Point F	0	0	10	5

**Tabel 4.7:** Trafiekmatrix met de kleinste link belasting.



**Figuur 4.7:** Simulatie op basis van tabel 4.7

Het is dus ook mogelijk om op deze manier naar een worst-case scenario te gaan. Voor dit probleem blijkt dat er meerdere worst-case scenario's zijn, namelijk vier met elk een totale belasting van 220. Figuur 4.6 is hier een voorbeeld van. Deze vier gevallen hebben dezelfde algemene trafiekmatrix namelijk tabel 4.8. Het enige verschil is hier dus de mapping van de injection points op reële nodes. De combinaties van nodes die hiervoor zorgen zijn (D,F), (D, G), (E,F) en (E,G).

Dienst X	Node D	Node E	Node F	Node G
Injection Point 1	0	0	10	5
Injection Point 2	25	15	0	0

**Tabel 4.8:** De trafiekmatrix waarvan bepaalde mappings van injection points zorgen voor maximale belasting.

### 4.3 Toepassing

Het zoeken naar een configuratie waarbij bepaalde objectieven vooropgesteld worden, is een interessant gegeven voor de upgradeplanning. Zoals hierboven aangetoond kan de link belasting van het netwerk enorm gereduceerd worden door *content* dichterbij de gebruikers te brengen. In dit upgradeplanning zal dus een afweging gemaakt moeten worden van de kost van het introduceren van een *injection points* versus de kost om een bepaalde capaciteit te garanderen tussen nodes.

Echter het simuleren van alle mogelijkheden is onbegonnen werk voor het aantal nodes in het Proximus-netwerk. Voor twee *injection points* zou dit al neer komen op  $2^{600} * \binom{600}{2}$  mogelijkheden. Er zullen dus manieren gevonden moeten worden om op voorhand het aantal mogelijkheden sterk te reduceren en zeer specifieke matrices uit te werken.

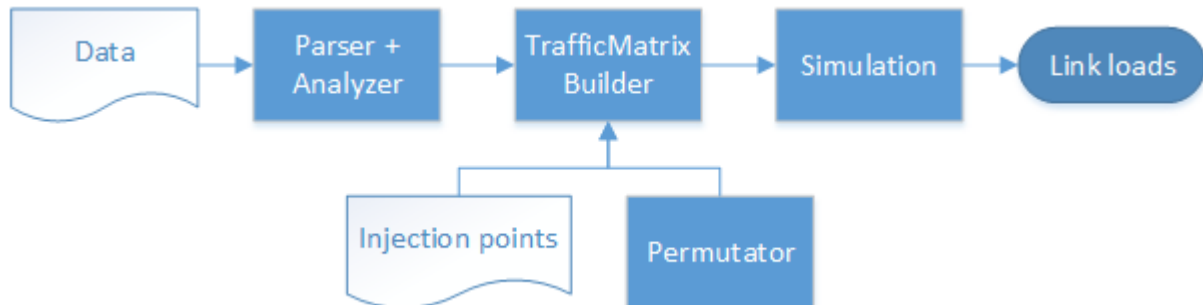
Het uitsluiten op basis van hop-afstand is een goede start om het aantal mogelijkheden te reduceren. Dit kan gebeuren door eerst voor alle nodes de hop-afstand te berekenen naar alle mogelijke *injection points*. Bij de selectie van een *injection point* voor een bepaalde node, gaat de voorkeur naar de *injection points* met een zo laag mogelijke hop-afstand.

Eveneens is het een optie om het aantal nodes te reduceren tot bijvoorbeeld enkel de *core* en *distribution* lagen<sup>2</sup>. Dus in plaats van alle 600 nodes te beschouwen, kan het aantal mogelijkheden reeds gereduceerd worden door enkel de nodes in de *core* en *distribution* lagen te beschouwen. Zelf dan zullen er nog te veel mogelijkheden zijn om allemaal te simuleren.

In de tool, hoofdstuk 3, is het mogelijk om de *injection points* afwisselend toe te wijzen en een op voorhand gedefinieerde lijst te gebruiken voor de mapping op reële nodes. Het zoeken naar de meest optimale oplossing voor het Proximus-netwerk is een optimalisatie-probleem

<sup>2</sup>Core, distribution en access lagen werden reeds besproken in hoofdstuk 2

op zich en valt buiten de scope van deze masterproef. Het genereren van trafiekmatrices is geen enkel probleem en deze kunnen, gegeven een topologie, ook gesimuleerd worden.



**Figuur 4.8:** De applicationflow voor het generen van trafiekmatrices

In figuur 4.8 is een overzicht te vinden van de applicationflow voor het genereren van trafiekmatrices. De eerste twee stappen werden reeds besproken in hoofdstuk 3. In de derde stap worden de trafiekmatrices gegenereerd. Dit gebeurt op basis van op voorhand gedefinieerde *injection points*, hiervoor is een configuratie-bestand voorzien. De permutator beslist hoe de *injection points* toegewezen worden aan de nodes, dit kan afwisselend of alle mogelijkheden. Vervolgens wordt deze output gebruikt in de netwerksimulatiebibliotheek om de routes en link loads te berekenen binnen het netwerk.

## 4.4 Conclusie

In dit hoofdstuk werd een methodologie voorgesteld voor het opbouwen van trafiekmatrices op basis van de bandbreedtevereisten voor een bepaalde dienst in het Proximus netwerk. Deze trafiekmatrices gebruiken injection points, initiële virtuele nodes binnen het netwerk die als bron dienen van alle trafiek. De virtuele injection points kunnen op reële nodes gemapped worden. Het zoeken naar een optimale oplossing voor de schaal van het Proximus netwerk, is een optimalisatieprobleem dat niet verder behandeld werd.

## Hoofdstuk 5

# Analyse van trends

Op basis van de output van de tool kunnen verschillende analyses gemaakt worden. In dit hoofdstuk worden deze analyses besproken. Deze analyses geven een algemeen beeld van de trafiek binnen het netwerk van Proximus. Opnieuw kunnen de verschillende aggregatieniveaus besproken in sectie 3.2 gebruikt worden.

Voor de upgradeplanning is het heel interessant om te weten wat de *peak-throughput* is. Dit is namelijk het moment waarop het netwerk het meest belast is. Als een node tijdens dit moment geen problemen heeft, dan zal hij met alle andere momenten ook geen problemen hebben.

Een andere vraag is, of er een verschil is qua trafiek tussen de verschillende SIMBA-zones<sup>1</sup>. Hiervoor worden de verschillende aggregatieniveaus ook nog eens per zone uitgevoerd, daarna worden ze vergeleken. Het is mogelijk om een dagelijks en maandelijks patroon op te stellen. Daarnaast worden de contributies per klant berekend. Opnieuw is hier de vraag of er een verschil is per zone.

De data kan ook op verschillende manieren in de tijd voorgesteld worden. In dit hoofdstuk worden de dagelijks- en maandelijks verbruikspatronen besproken. Het tijdsinterval voor het dagelijks verbruikspatroon is vijf minuten, dit kan in de tool aangepast worden om bijvoorbeeld aggregaties per 30 minuten te bekomen. Wanneer er een maandelijks verbruikspatroon opgesteld wordt, dan vertrekt de tool van het eerste chronologisch beschikbare inputbestand. Alle aggregaties die kunnen gemaakt worden op basis van dit bestand, kunnen voor de hele maand gemaakt worden. Concreet betekent dit wanneer

---

<sup>1</sup>Zie sectie 3.2

bijvoorbeeld op nodeniveau een node bijgevoegd wordt in het midden van de maand, deze niet in het maandoverzicht zal worden opgenomen.

Verder wordt de informatie ook nog eens geografisch voorgesteld op basis van shape-files. Deze weergave toont de evolutie doorheen de dag. Op deze manier kan een overzicht gekregen worden van de belasting op het netwerk doorheen de dag. Deze evolutie kan ook per week, per maand, ....

## 5.1 Analyseren van trafiek

Er zijn verschillende diensten die door Proximus aangeboden worden. In dit hoofdstuk wordt specifiek gewerkt met FIA\_RES en FIA\_PRO. Elke analyse die hier gemaakt wordt kan analoog ook voor de andere diensten en de verschillende aggregators gemaakt worden. Een dienst, zoals aangegeven in hoofdstuk 3, is niet meer dan een verzameling van vlan's die via een configuratiebestand ingesteld kunnen worden.

Eerst worden, in deze sectie, de volgende aggregatieniveaus besproken network, zone, node en ESS voor FIA\_RES. Dit gebeurt op basis van de peak-throughput over de hele maand. Vervolgens wordt er gekeken wanneer deze peak-throughput zich juist voordoet. Dit gebeurt door middel van histogrammen. Ten slotte wordt de vergelijking gemaakt met FIA\_PRO. Er wordt specifiek met egress-waarden gewerkt. Het is ook mogelijk om deze analyses met de ingress-waarden te maken.

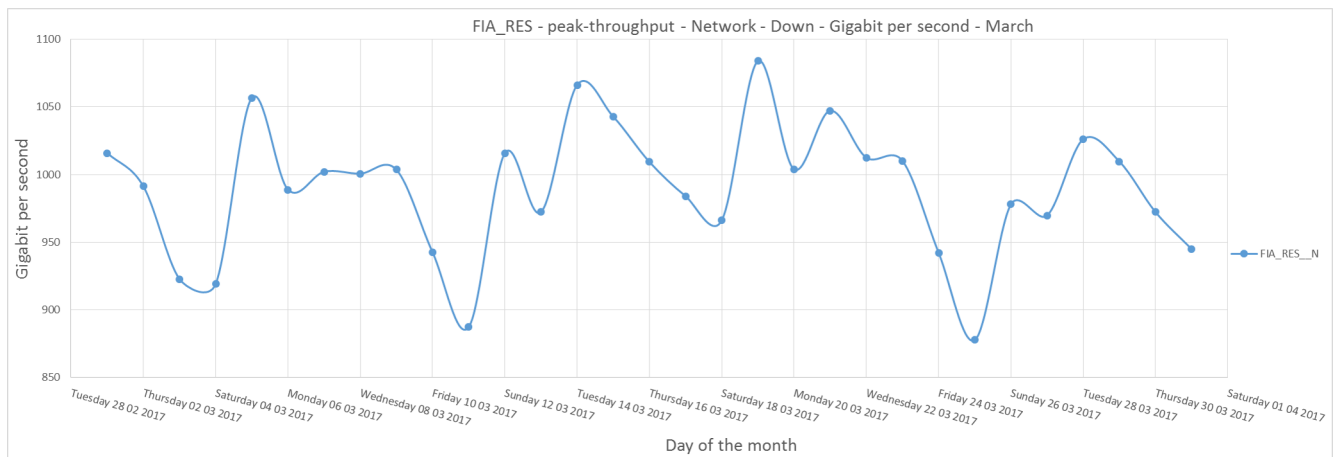
### 5.1.1 Dienst FIA\_RES

#### Network

Figuur 5.1 geeft de *peak-throughput* voor de hele maand maart. Op deze grafiek is te zien dat zaterdagen en vrijdagen een duidelijk dal vertonen. Verder valt de piek van deze grafiek op zondag, 17 maart met 1084 Gbps.

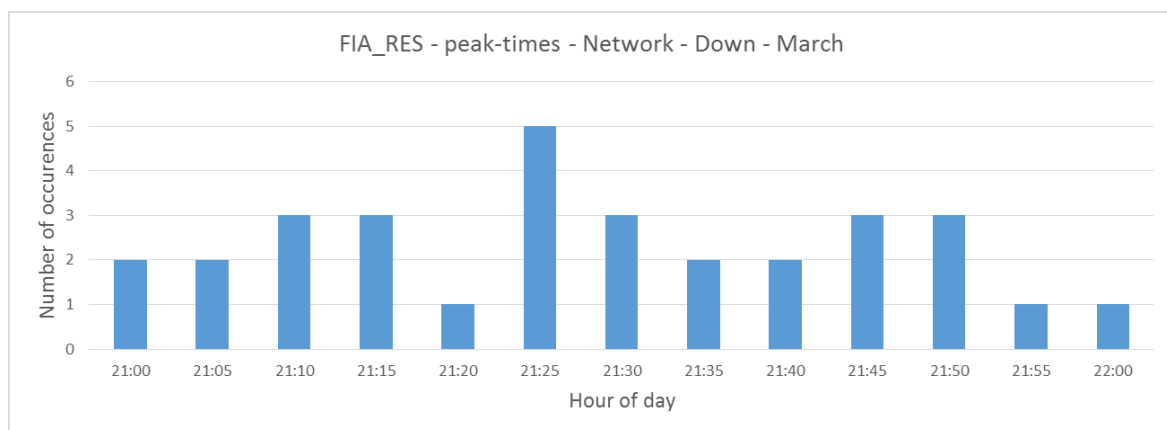
Een andere vraag is, op welk tijdstip krijgt het netwerk het meeste trafiek te verwerken. Met andere woorden wanneer gebeurt de peak-throughput. In figuur 5.2 wordt een overzicht gegeven van de peak-times voor maart. Het verband met figuur 5.1 is dat voor elk punt op die grafiek het overeenkomstige tijdstip waarop die throughput gebeurt, gebruikt wordt





**Figuur 5.1:** De peak-throughput voor de dienst FIA\_RES in maart.

om een frequentietabel op te stellen. De peak-times voor FIA\_RES liggen tussen 21:00 en 22:00 uur. Tussen deze twee uren moet het netwerk, het meeste trafiek verwerken.

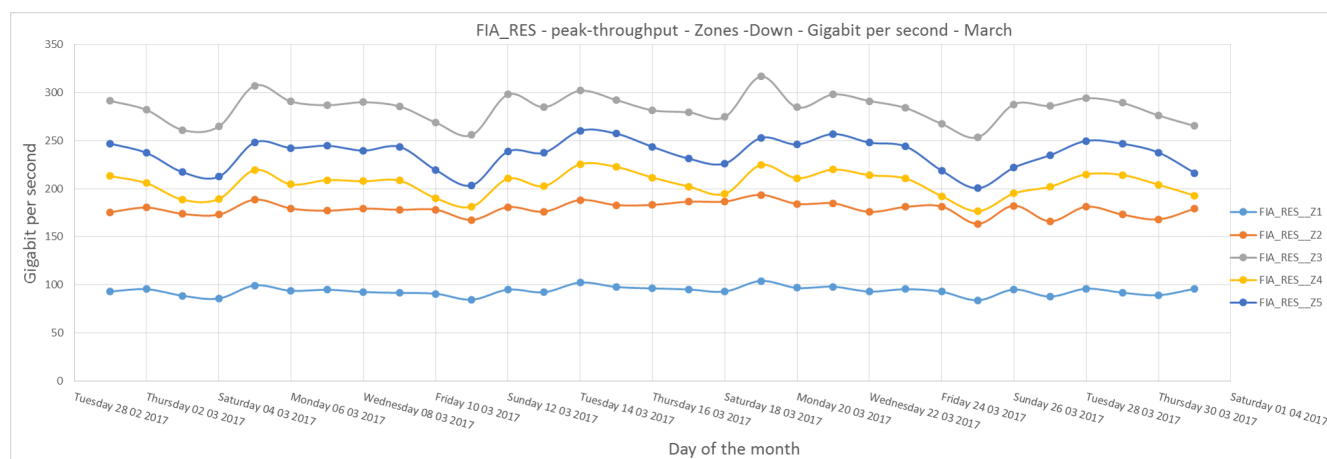


**Figuur 5.2:** De peak-times voor de dienst FIA\_RES in maart.

## Zone

De grafiek van figuur 5.1 kan nu verder opgesplitst worden in de verschillende SIMBA-zones. De totale throughput voor het hele netwerk zal natuurlijk gelijk blijven, echter de vraag is of de verschillende zones zich gelijk gedragen of niet. In figuur 5.3 wordt een overzicht van deze verdeling gegeven. In deze figuur is duidelijk op te merken dat er substantiële verschillen zijn tussen de zones. Merk ook op dat er elke zaterdag een duidelijk dipje te

zien is. Hoewel voor het hele netwerk de piek op zondag 17 maart viel, is dit voor de zones niet zo. Z4 en Z5 hebben hun piek op dinsdag 14 maart.



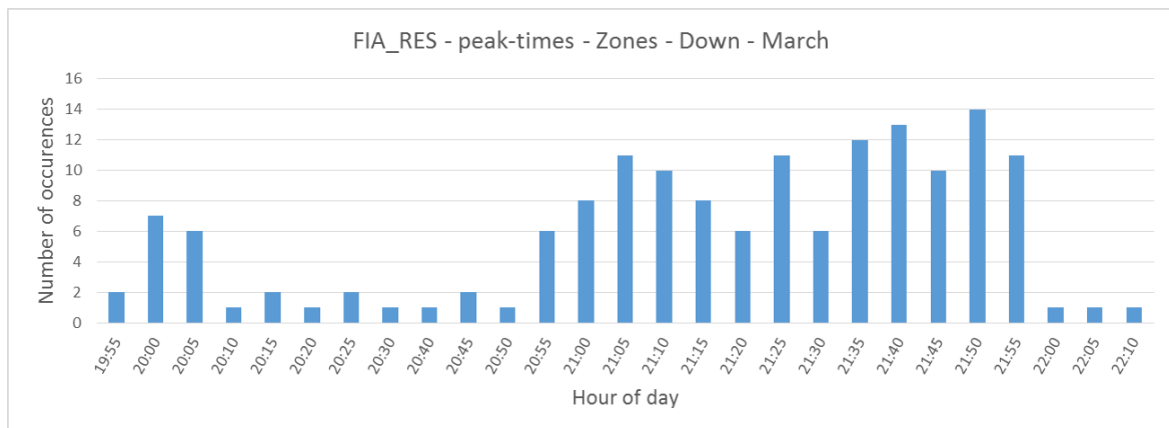
**Figuur 5.3:** De *peak-throughput* verdeeld over de verschillende SIMBA-zones voor de dienst FIA\_RES in maart.

Als op basis van deze data een gemiddelde *peak-throughput* per zone voor de maand maart gemaakt wordt, is het resultaat tabel 5.1. Zoals ook op te merken in de grafiek verbruikt Z1 minder dan de andere zones. Het verschil tussen Z2 en Z3 is ook substantieel.

Aggregatie-niveau	Gigabit per seconde
Network	989,2
Z1	93,7
Z2	179,1
Z3	283,6
Z4	205,2
Z5	236,3

**Tabel 5.1:** Gemiddelde *peak-throughput* in maart voor FIA\_RES, down.

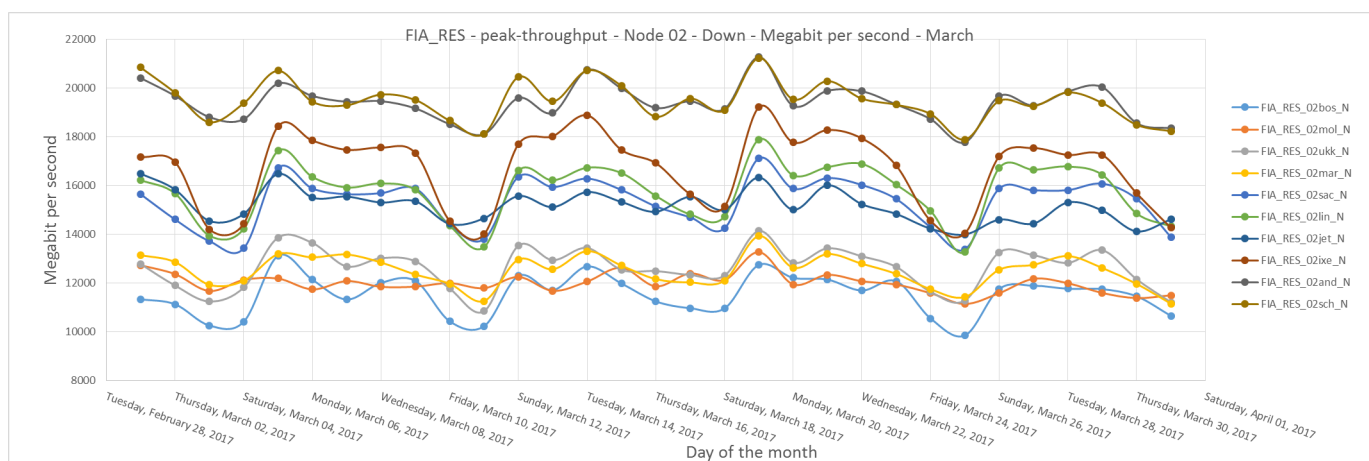
Opnieuw is het mogelijk om naar de peak-times van de verschillende zones te kijken. Uit figuur 5.4 kan men afleiden dat de peak-times voor de zones anders kunnen liggen dan die van het netwerk. Echter ligt het grootste aantal peak-times nog altijd tussen 21:00 en 22:00 uur.



**Figuur 5.4:** De peak-times van de verschillende zones voor de dienst FIA\_RES in maart.

## Node

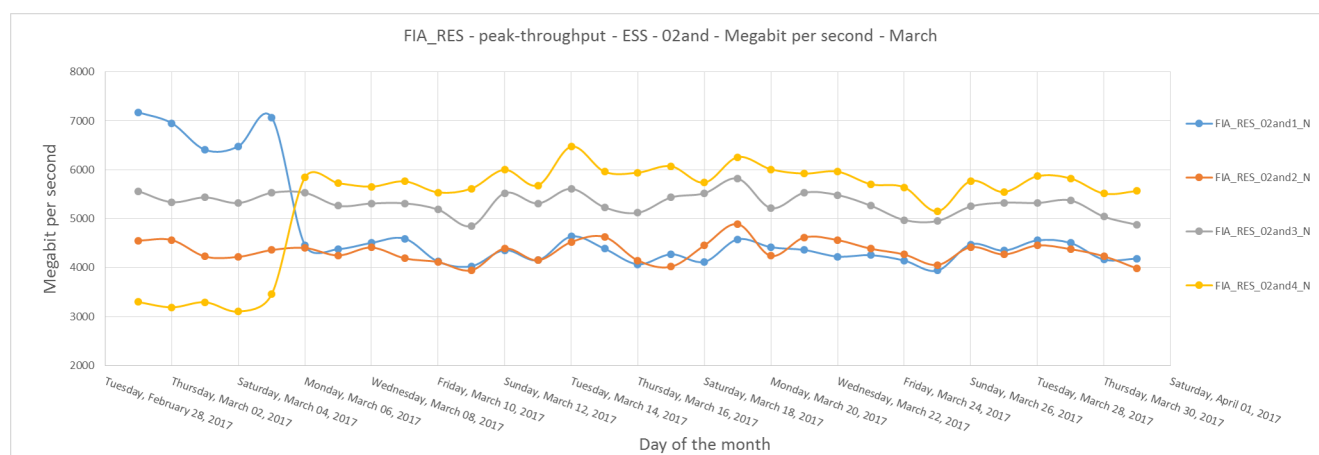
Op nodeniveau kunnen dezelfde grafieken gemaakt worden. Omdat er een 600-tal nodes zijn is het onoverzichtelijk om deze allemaal samen op dezelfde grafiek te zetten. De phonezone 02 heeft in totaal 43 nodes, hiervan worden er 10 als voorbeeld genomen. In figuur 5.5 staat de *peak-throughput* van deze nodes uitgezet over de maand maart. Opnieuw kan men vaststellen dat er verschillen zijn tussen de nodes. De piek van zondag 19 maart is opnieuw te zien in deze grafiek. Vrijdag en zaterdag vormen opnieuw een dal doorheen de hele maand maart.



**Figuur 5.5:** De *peak-throughput* bij een aantal nodes van de 02 phonezone voor de dienst FIA\_RES in maart.

## ESS

Figuur 5.6 geeft een overzicht van de trafiek die door verschillende ESS'en van de 02and-node verwerkt worden in de maand maart. Op 5 maart is er een duidelijke overdracht te zien tussen ESS 1 en 4. Wanneer een ESS te veel belast wordt, dan worden een deel van de klanten verplaatst naar een ander ESS.



**Figuur 5.6:** De *peak-throughput* bij de verschillende ESS'en van de 02and-node voor de dienst FIA\_RES in maart.

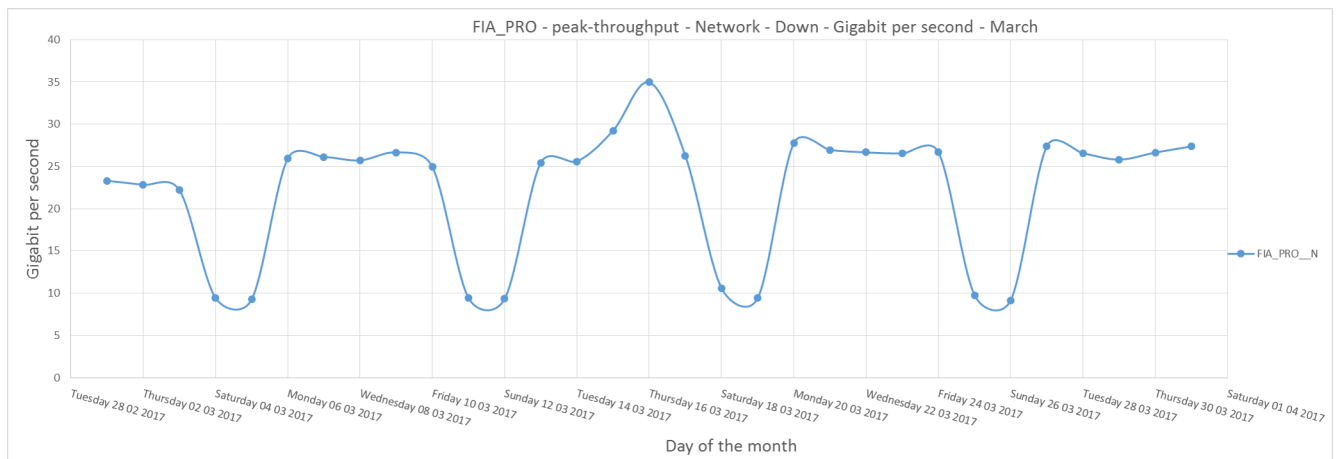
### 5.1.2 Dienst FIA\_PRO

In deze subsectie zal de vergelijking met FIA\_PRO gemaakt worden, op basis van dezelfde soort grafieken uit de vorige subsectie.

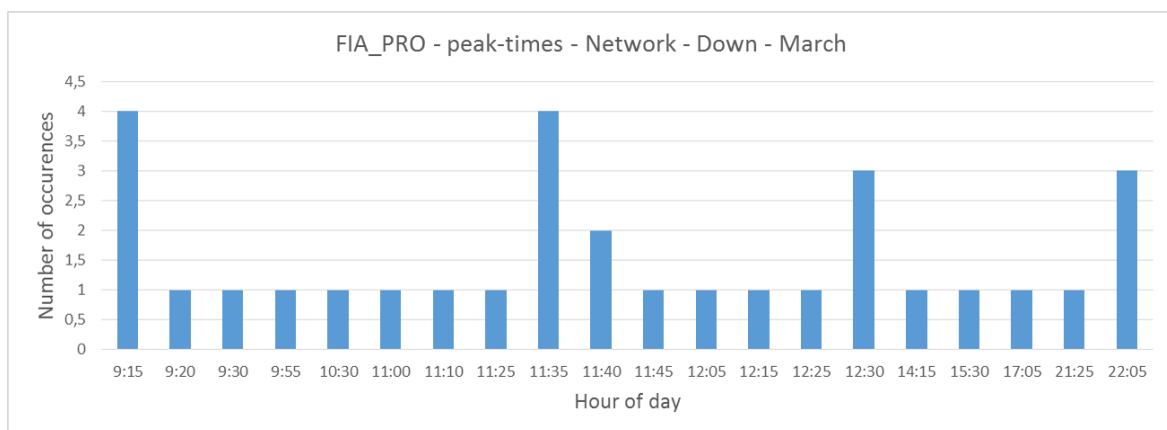
#### Network

In figuur 5.7 staat de *peak-throughput* voor de hele maand maart. Hier is een duidelijk patroon te zien. Tijdens werkdagen is er de *peak-throughput* gemiddeld 26,4 Gbps terwijl tijdens de weekends dit 9,56 Gbps is. Dit verband is niet terug te vinden in dezelfde grafiek voor FIA\_RES.

De peak-times voor FIA\_PRO liggen volledig anders dan die voor FIA\_RES. De meeste peak-times zijn te vinden tussen 09:15 en 11:40 uur. Op basis van dit resultaat kan men verwachten dat het *daily pattern* voor beide diensten anders zal zijn.



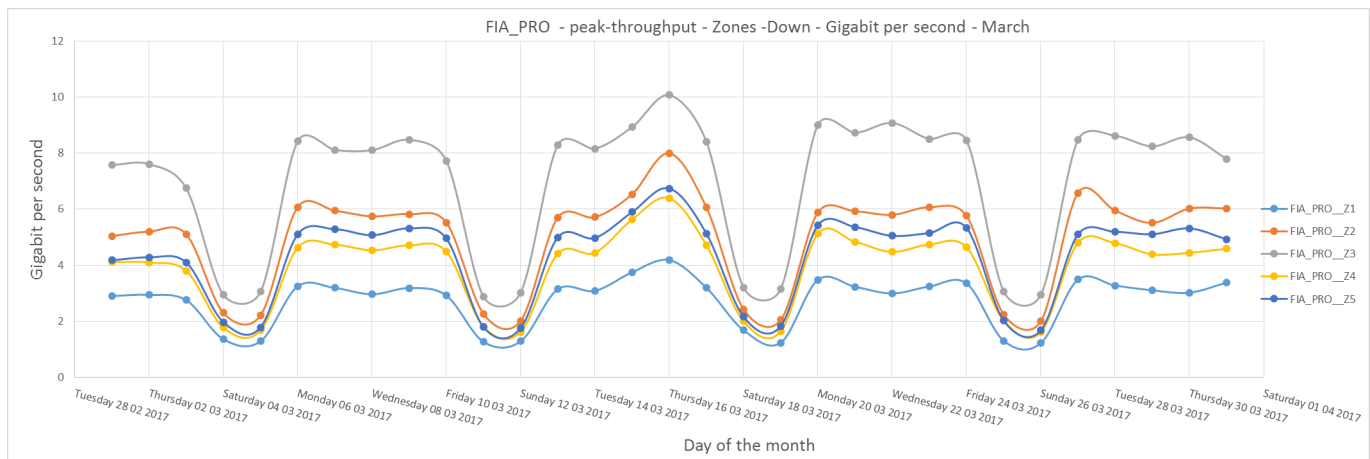
**Figuur 5.7:** De *peak-throughput* voor de dienst FIA\_PRO in maart.



**Figuur 5.8:** De *peak-times* voor de dienst FIA\_PRO in maart.

## Zone

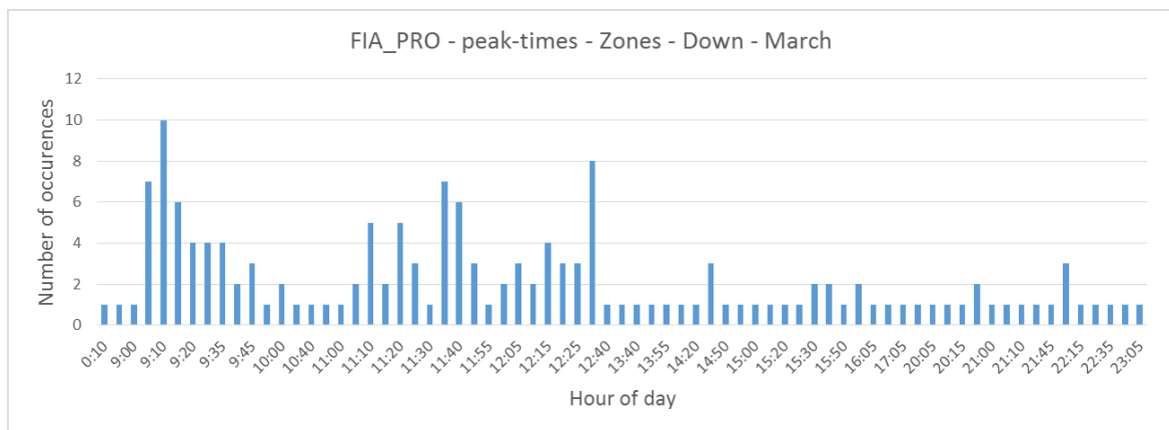
De informatie uit figuur 5.7 kan opnieuw verdeeld worden over de verschillende SIMBA-zones, dit geeft als resultaat figuur 5.9. Opnieuw moet Z3 het meeste trafiek verwerken, terwijl Z1 onderaan de grafiek staat. Dit was ook te zien in figuur 5.3 voor FIA\_RES. Voor FIA\_PRO liggen Z2, Z3 en Z5 veel dichtter bij elkaar dan FIA\_RES. De gemiddelde *peak-throughput* per zone is terug te vinden in tabel 5.2. Hierin worden bovenstaande bevindingen bevestigd.



**Figuur 5.9:** De *peak-throughput* verdeeld over de verschillende SIMBA-zones voor de dienst FIA\_PRO in maart.

Aggregatie-niveau	Gigabit per seconde
Network	22,1
Z1	2,7
Z2	5
Z3	7
Z4	4
Z5	4,3

**Tabel 5.2:** Gemiddelde *peak-throughput* in maart voor FIA\_PRO, down.

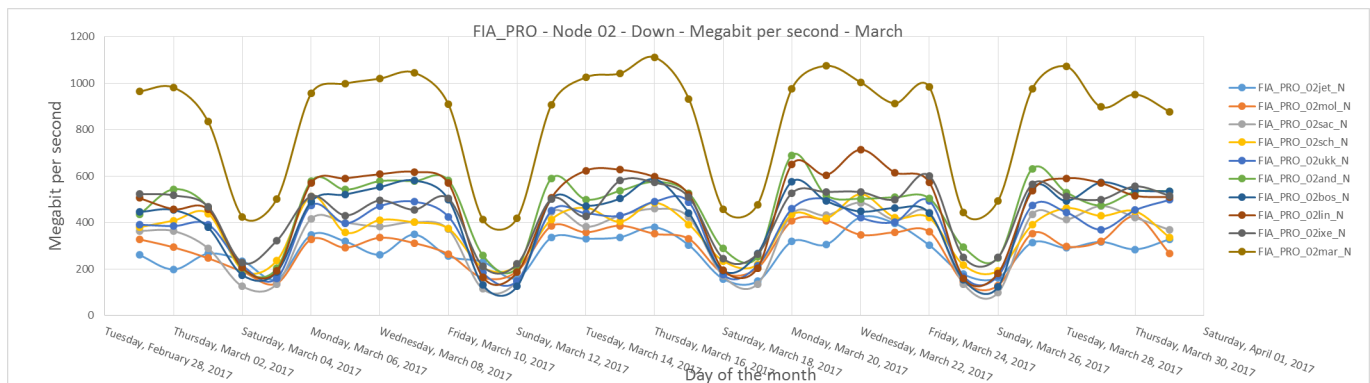


**Figuur 5.10:** De peak-times van de verschillende zones voor de dienst FIA\_PRO in maart.

### Node

In figuur 5.11 wordt het nodeniveau bekeken. Hierin staat terug een overzicht van een aantal nodes uit de 02-phonezone. Opnieuw is er een verschil tussen de verschillende

nodes, de trend is echter wel dezelfde.



**Figuur 5.11:** De *peak-throughput* bij een aantal nodes van de 02-phonezone voor de dienst FIA\_PRO in maart.

## 5.2 Analyseren van *daily pattern*

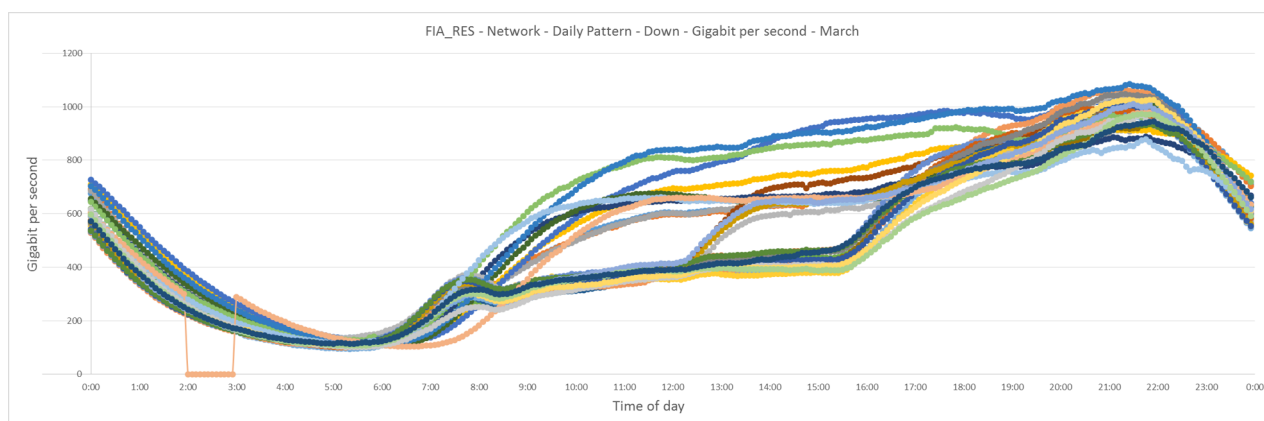
Naast maandoorzichten van de diensten is het mogelijk om de dagelijkse verbruikspatronen te maken. Hierbij wordt om de vijf minuten een punt toegevoegd aan de grafiek. Opnieuw kan dit voor alle aggregatieniveaus besproken in 3.2 en diensten.

In deze sectie wordt eerst FIA\_RES besproken daarna wordt, analoog met vorige sectie, de vergelijking gemaakt met FIA\_PRO. De *daily pattern* wordt voor het netwerk-niveau besproken.

### 5.2.1 Analyseren van *daily pattern* voor FIA\_RES

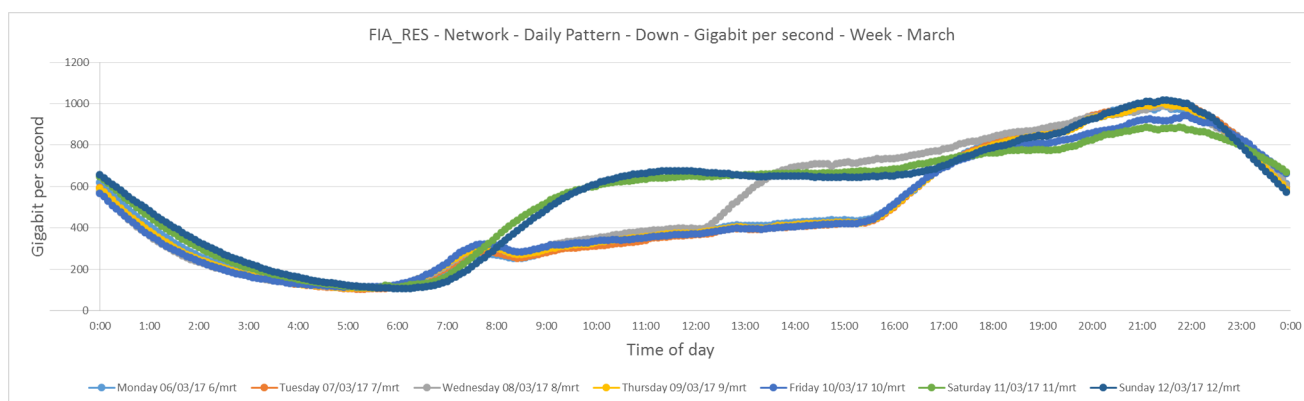
Figuur 5.12 toont het *daily pattern* voor FIA\_RES in de maand maart. Elke lijn stelt dus één dag voor. Uit deze figuur kan men afleiden dat het netwerk het minst belast is tussen 02:00 en 06:00 uur. Verder valt ook op te merken dat de patronen verschillen per dag. Van 02:00 tot 03:00 zijn er geen metingen op 26/03, dit verklaart dus waarom er op de grafiek 0-waarden te zien zijn. Om een duidelijk beeld te geven van hoe deze patronen verschillen zal deze grafiek opgesplitst worden in meerdere subgrafieken.

In figuur 5.13 staan zeven dagen aan aggregaties uitgestippeld. Er zijn drie opvallende trends te merken. Voor zaterdag en zondag is er een duidelijke stijging van throughput rond



**Figuur 5.12:** Het *daily pattern* voor de dienst FIA\_RES in maart.

08:00-09:00 uur tegenover de andere dagen. Woensdag is er ook een stijging te zien, tijdens de middag. Deze drie trends worden nu verder geanalyseerd in de volgende grafieken.

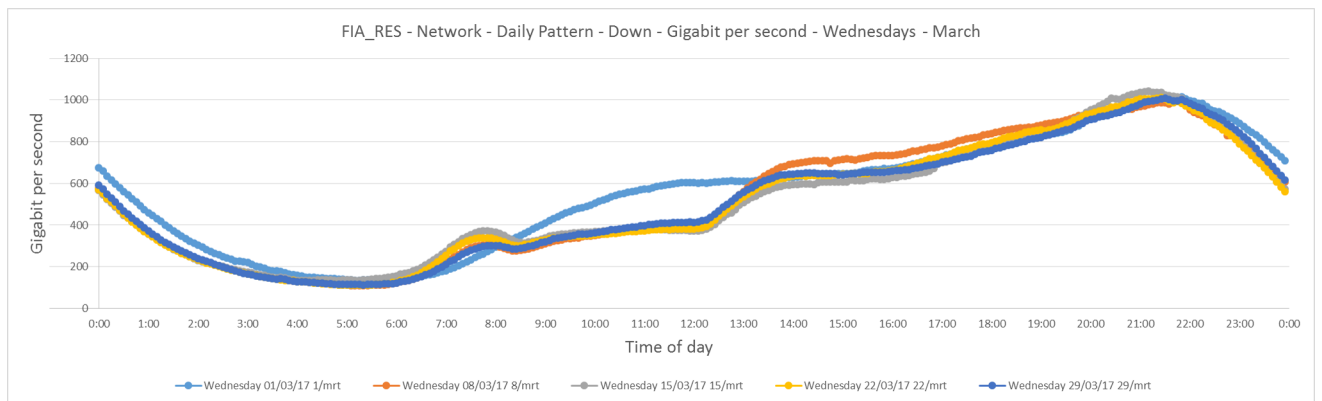


**Figuur 5.13:** Het *daily pattern* voor week 10 van de dienst FIA\_RES in maart.

Zijn de trends in figuur 5.13 geldig voor alle woensdagen, weekends en weekdagen? In figuur 5.14 staan alle woensdagen op één grafiek. Voor vier van de vijf woensdagen wordt deze trend dus bevestigd. Een mogelijke verklaring dat 1 maart afwijkt, is omdat het dan nog Krokusvakantie was. Dit is echter moeilijk te valideren, zou zouden meerdere woensdagen uit Krokusvakanties met elkaar vergeleken moeten worden. Zoveel data wordt er historisch niet bij gehouden door Proximus.

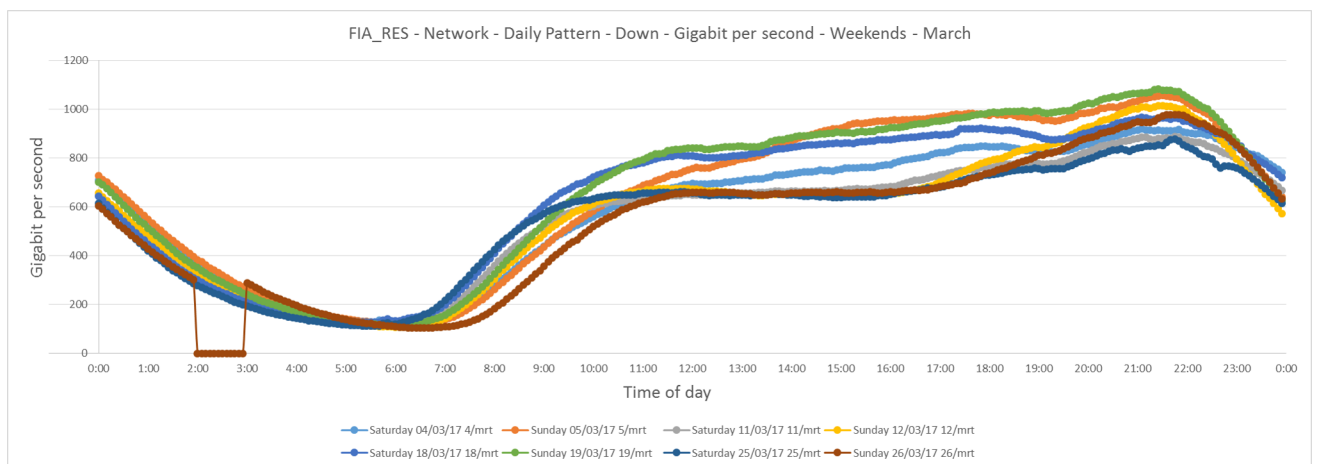
Voor de weekends is er een variatie van ongeveer 300 Gbps van 12:00 tot 22:00 uur, zie figuur 5.15. Tussen 07:00 en 11:00 uur is er een initiële stijging, waarna er een verdere stijging is naar de piek van de dag toe. Deze variatie is niet terug te vinden bij de weekdagen





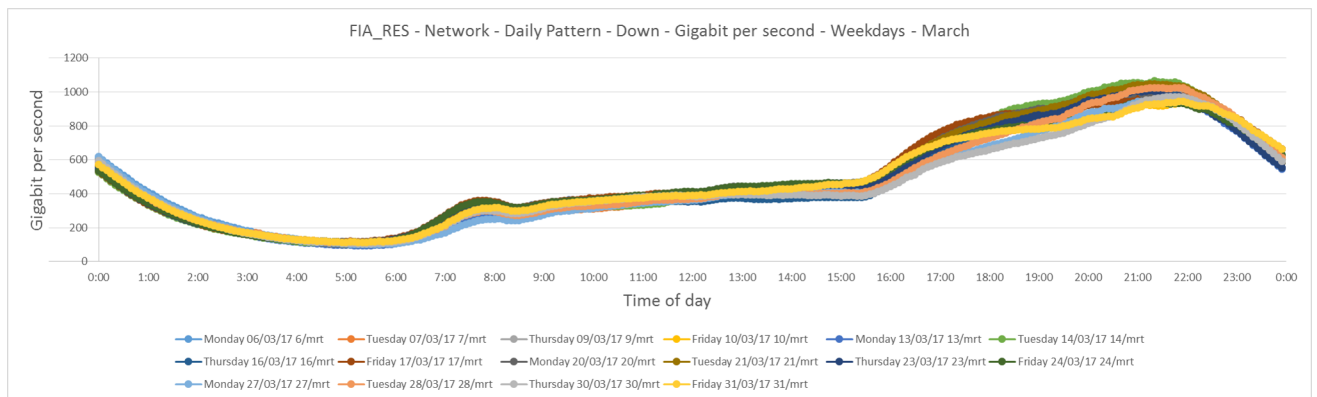
**Figuur 5.14:** Het *daily pattern* voor alle woensdagen van de dienst FIA\_RES in maart.

in figuur 5.16. Voor de weekdays is er een zeer duidelijke trend te zien die een beetje speling heeft in de late namiddag.

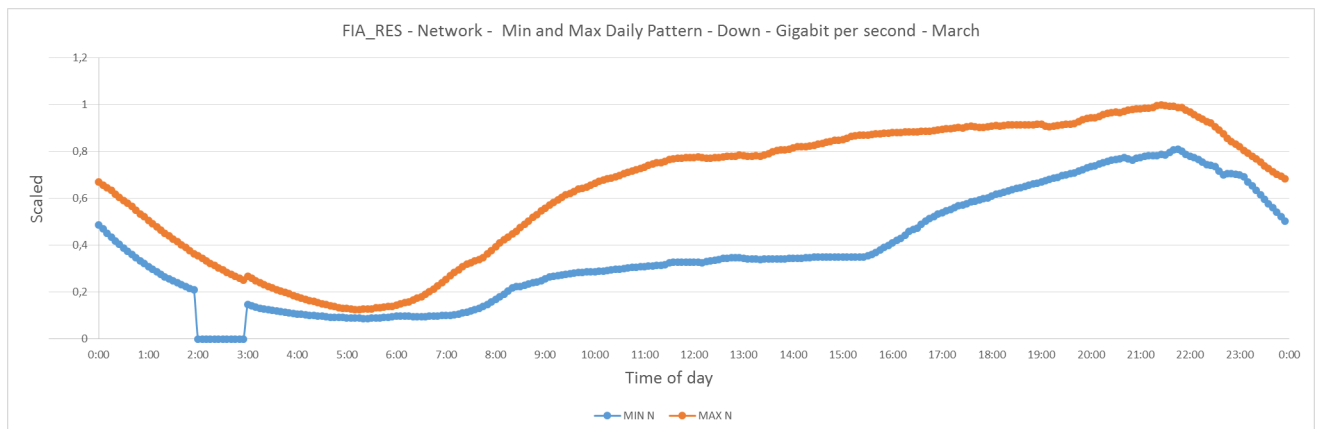


**Figuur 5.15:** Het *daily pattern* voor alle weekends van de dienst FIA\_RES in maart.

Nog een heel interessant gegeven is de verhouding tussen het minimum en maximum *daily pattern*. Hierbij wordt dus gekeken naar de marge tussen minimum en maximumbelasting. Om dit op een duidelijke manier weer te geven worden de gegevens geschaald van 0 tot 1, dit is te zien in figuur 5.17. Merk op dat er dus een gemiddelde marge is van 27% over de hele dag. Deze gemiddelde marge loopt op tot 45% tussen 10:00 en 16:00 uur.



**Figuur 5.16:** Het *daily pattern* voor alle weekdays van de dienst FIA\_RES in maart.

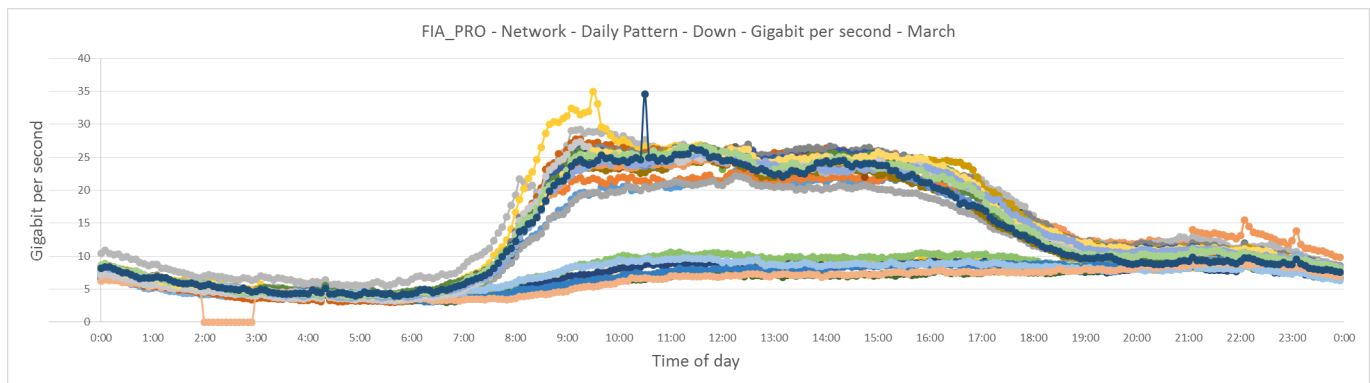


**Figuur 5.17:** Het *daily pattern* voor alle weekdays van de dienst FIA\_RES in maart.

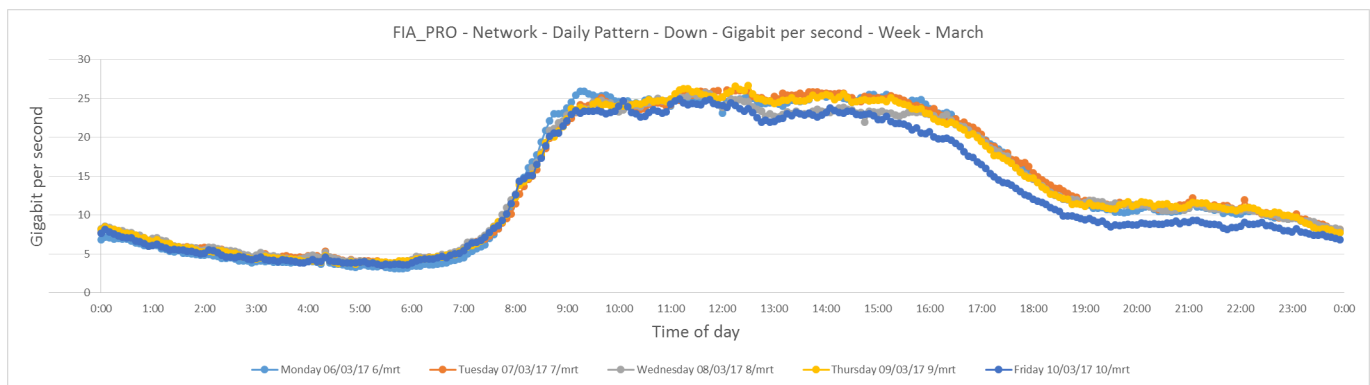
## 5.2.2 Analyseren van *daily pattern* voor FIA\_PRO

Zoals eerder vermeldt zijn de peak-times voor FIA\_RES en FIA\_PRO verschillend. Het *daily pattern* zal daarom verschillen van dat van FIA\_RES. In figuur 5.18 staat de *daily pattern* voor alle dagen van maart. Tussen 01:00 en 06:00 is er opnieuw weinig trafiek voor FIA\_PRO. Voorts zijn er twee duidelijke trends te merken. De grafiek zal dus opgesplitst worden in werkdagen en weekends.

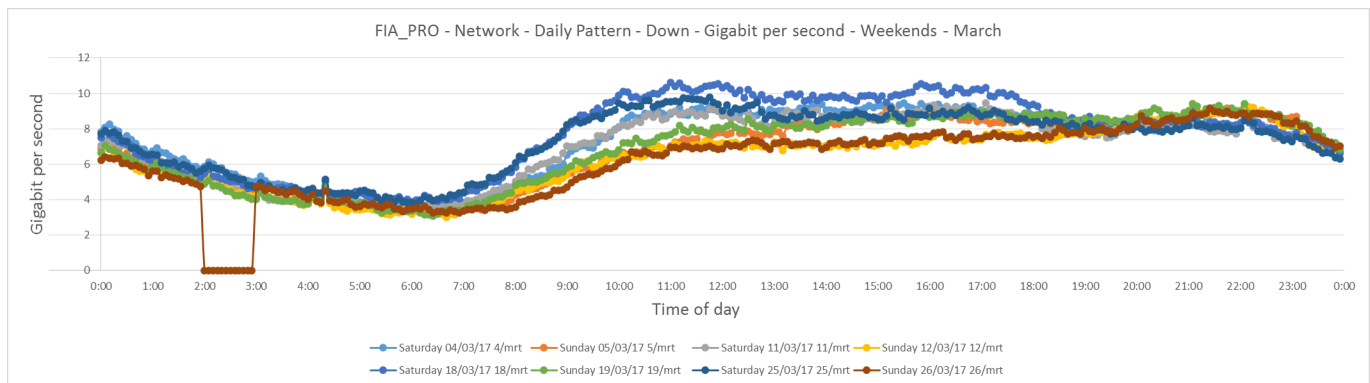
In figuur 5.19 staat een typische week voor FIA\_PRO. Op deze grafiek is een duidelijke trend te zien en er is weinig variatie tussen de verschillende werkdagen. Figuur 5.20 geeft de weekends weer. Opnieuw met een heel duidelijk verloop. Er zijn dus twee verschillende verlopen voor FIA\_PRO, namelijk werkdagen en weekends.



**Figuur 5.18:** Het *daily pattern* voor de dienst FIA\_PRO in maart.



**Figuur 5.19:** Het *daily pattern* van een typisch week voor de dienst FIA\_PRO in maart.



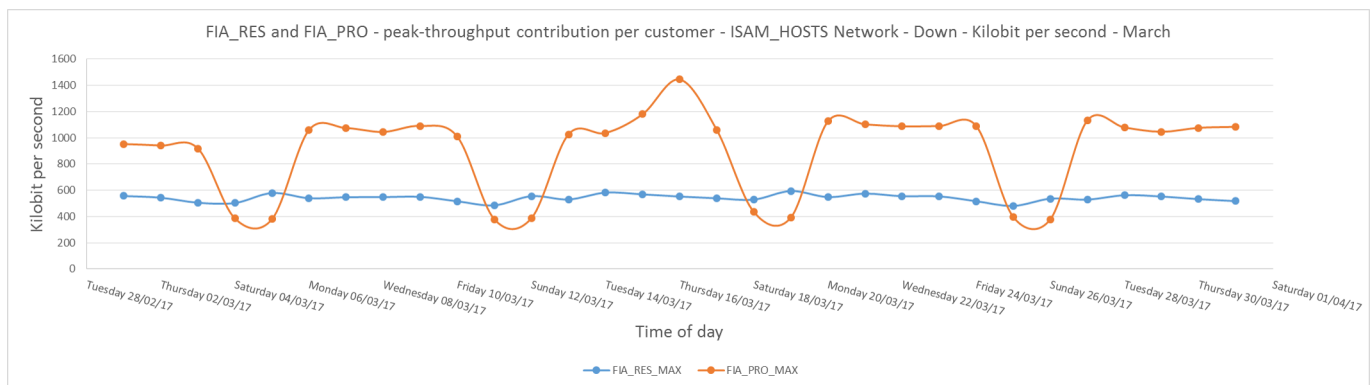
**Figuur 5.20:** Het *daily pattern* voor alle weekends van de dienst FIA\_PRO in maart.

Er is dus een duidelijk verschil met FIA\_RES. Het verschil in peak-times wordt nogmaals bevestigd in het *daily pattern*.

## 5.3 Contributie per klant

Op basis van de klanteninformatie, besproken in hoofdstuk 2, is het mogelijk om per aggregatortype een gemiddelde contributie te bepalen. Op deze manier kan men vergelijken of klanten zich in de verschillende SIMBA-zones zich anders gedragen. Verder kan men ook de verschillende nodes onderling met elkaar vergelijken.

Figuur 5.21 geeft de gemiddelde contributie per klant voor zowel FIA\_RES en FIA\_PRO voor ISAM\_HOSTS. De tool vindt voor FIA\_RES 1.582.811 klanten en voor FIA\_PRO 20.344. Er zijn dus veel minder FIA\_PRO klanten die gemiddeld meer verbruiken dan FIA\_RES klanten.



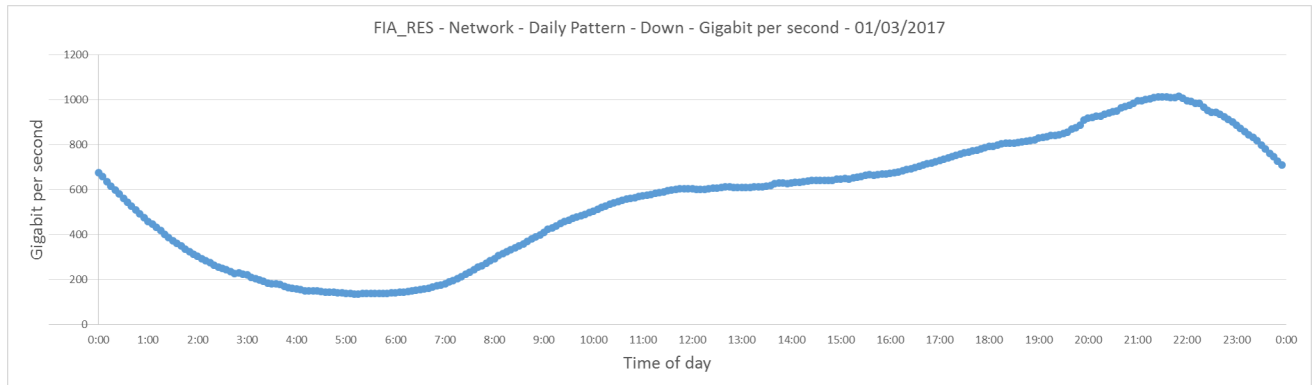
**Figuur 5.21:** De peak-throughput contribution per customer voor zowel FIA\_RES en FIA\_PRO bij ISAM\_HOSTS.

## 5.4 Heatmaps

Het maken van deze heatmaps gebeurt op basis van de output van de tool besproken in hoofdstuk 3. Dit gebeurt in een apart Java Maven-project. Op basis van het *daily pattern* (sectie 5.2) op nodeniveau en de geografische informatie (sectie 2.3), is het mogelijk om per node een evolutie doorheen de dag te tonen. Op deze manier kan de belasting van het netwerk gevisualiseerd worden op de kaart van België. De tool maakt per vijf minuten een afbeelding. Deze bestanden worden vervolgens omgezet in een GIF-bestand. Het GIF-bestand toont per 500 ms, één afbeelding.

Om een evolutie te kunnen maken worden de gegevens geschaald en dan over groepen verdeeld. Deze groepen krijgen elk een kleur toegewezen. Er wordt geschaald op basis van

de maximum throughput-waarde over alle nodes voor die dag. Hoewel er in deze sectie gewerkt wordt met het *daily pattern*, is het ook mogelijk om heatmaps te maken voor de hele week en maand.



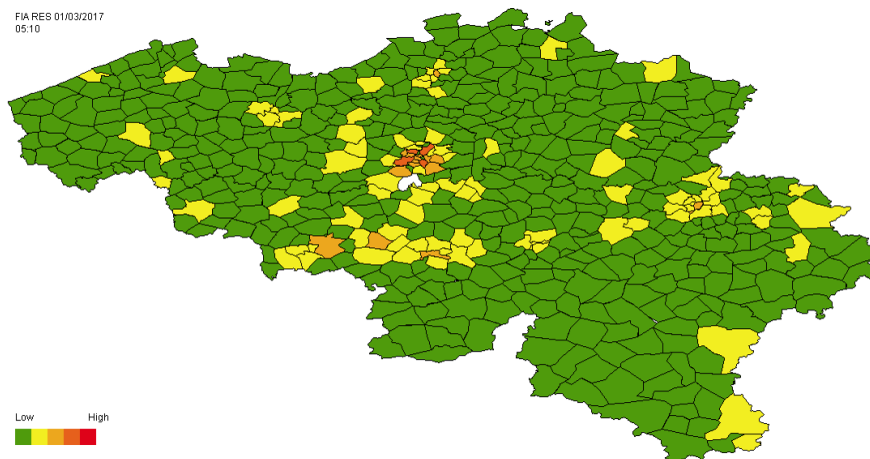
**Figuur 5.22:** Het *daily pattern* van de dienst FIA\_RES op 01/03.

Als voorbeeld wordt in deze sectie, de dienst FIA\_RES besproken op 01/03/2017. Om een duidelijker beeld te krijgen van het *daily pattern* voor die dag, zie figuur 5.22. In deze grafiek is duidelijk te zien dat de belasting een minimum kent rond 05:00 uur. Vervolgens is er een stijging tot de piek om 22:00 uur.

Om een goed idee te krijgen van de evolutie doorheen de dag, worden drie afbeeldingen besproken namelijk 05:10, 11:50, 21:55 uur. Om 05:10 uur wordt er 135 Gbps aan trafiek verwerkt binnen het netwerk. In figuur 5.23 staat de heatmap voor dit tijdstip. Er worden in dit geval vijf groepen gebruikt, waarbij groen staat voor de minste belasting en rood voor de meeste. Tijdens de nacht is er dus weinig activiteit te zien op de heatmap, wat ook al bevestigd werd door de grafiek in figuur 5.22.

De heatmap van 11:50 uur is te vinden in figuur 5.24. Hier zijn reeds enkele gebieden rood gekleurd. De belasting voor het hele netwerk is tegenover de vorige afbeelding verdriedubbeld. Er zijn ook meer gele gebieden te zien. De rode gebieden zijn de grotere steden zoals Antwerpen(521.216 inwoners), Gent(259.991), Charleroi(201.753), Luik(198.662), Brussel(177.863), Namen(111.376), Bergen(95.259) en Kortrijk(76.059). Merk dus op dat er een verband is tussen het aantal inwoners en de kleur van een bepaald gebied. Echter is dit verband nog niet te zien voor Brugge(118.504) en Leuven(100.651)<sup>2</sup>. Deze zijn beide

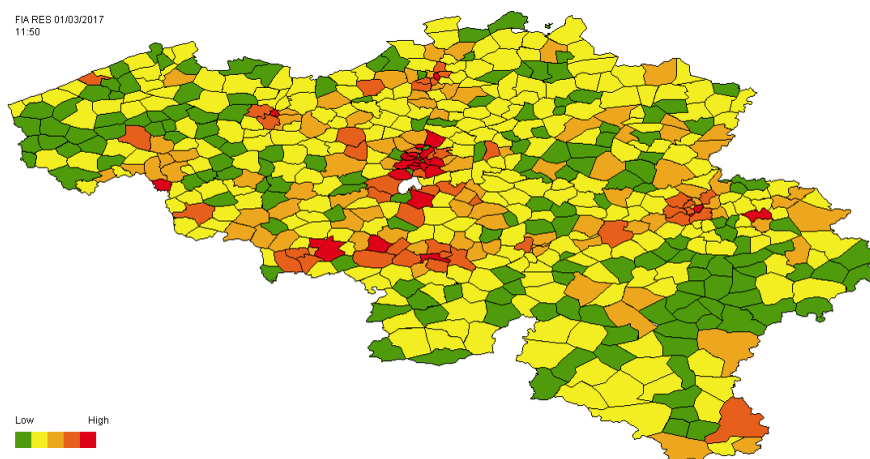
<sup>2</sup>Cijfers over bevolkingsaantallen; Situatie op 1-4-2017 : <http://www.ibz.rrn.fgov.be/nl/bevolking/statistieken-van-bevolking/>



**Figuur 5.23:** Heatmap voor de dienst FIA\_RES om 05:10 uur op 01/03/2017

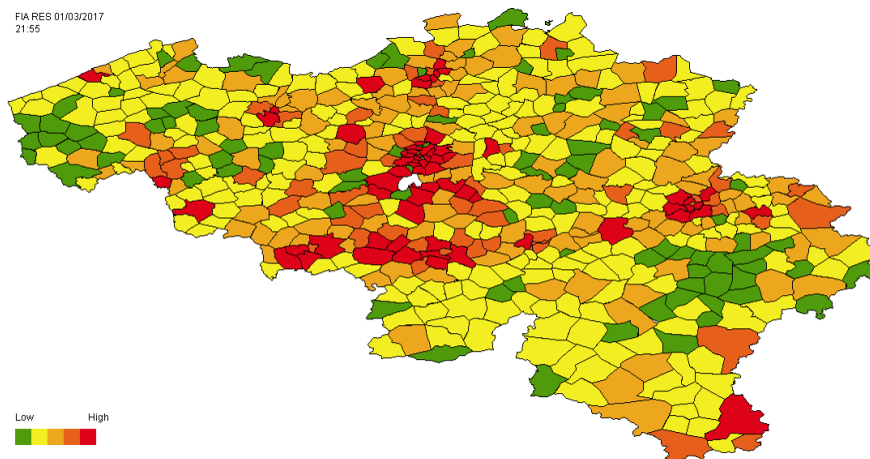
donkeroranje gekleurd, er is met andere woorden belasting maar niet evenveel als in de andere grootsteden.

Er zijn enkele gebieden wit-gekleurd in de heatmaps. Dit is geen fout in het programma. Sommige gebieden hebben zelf geen *backbone-equipment* en zijn dus afhankelijk van de *backbone-equipment* van een ander naburig gebied.



**Figuur 5.24:** Heatmap voor de dienst FIA\_RES om 11:50 uur op 01/03/2017

Om 21:55 uur, het piekmoment van de dag, is er de meeste belasting in het netwerk. Dit is duidelijk te zien in de heatmap, figuur 5.25. De grote populatiecentra van het land zijn allemaal rood gekleurd. Er zijn op dit moment nog enkele groen gekleurde gebieden.



**Figuur 5.25:** Heatmap voor de dienst FIA\_RES om 21:55 uur op 01/03/2017

## 5.5 Conclusie

Op basis van de uitvoer van de tool werden verschillende analyses gemaakt. Concreet werd FIA\_RES en FIA\_PRO met elkaar vergeleken voor zowel het *daily pattern* als *monthly pattern*. De verschillende SIMBA-zones werden ook met elkaar vergeleken. Verder werd ook de contributie per klant en heatmaps gerealiseerd.

## Hoofdstuk 6

### Conclusie

Het probleem van alsmaar toenemend IP-verkeer werd reeds uitgebreid besproken in hoofdstuk 1. Op korte termijn lijkt er geen einde in zicht voor de toenemende eisen op telecommunicatienetwerken. De verschillende technologieën die toegang geven tot het internet voor eindgebruikers, bieden steeds meer bandbreedte aan. Een belangrijk deel van het totale IP-verkeer is realtime entertainment [17, 12, 4]. De impact van 4K-streaming op de netwerken van telecomoperatoren mag dus zeker niet onderschat worden. Verder zal ook 5G voor extra trafiek zorgen.

Deze toenemende eisen zorgen er voor dat telecomoperatoren zorgvuldig moeten omgaan met hun upgradeplanning. In deze upgradeplanning moet rekening gehouden worden met de huidige en toekomstige trafiekstromen om een zo kwaliteitsvolle dienst aan te bieden. De upgradeplanning heeft uiteraard een beperkt budget. Indien het mogelijk is om de huidige infrastructuur beter te benutten, dan biedt dit nieuwe mogelijkheden voor de upgradeplanning.

In deze masterproef worden verschillende analyses gemaakt op basis van metingen binnen het Proximus netwerk. Deze metingen kunnen opgedeeld worden tussen bedrade en draadloze technologieën. Tevens is er informatie beschikbaar over de klantenaantallen en de geografische locaties van de nodes binnen het netwerk. In hoofdstuk 2 werd een uitgebreide uitleg over de data gegeven.

De analyses die gemaakt worden hebben betrekking tot de drie pijlers van het onderzoeksproject tussen UGent en Proximus, dat reeds besproken werd in sectie 1.4. In eerste instantie werd het concept van trafiekmatrices met *injection points* geïntroduceerd. Met trafiekmatrices is het mogelijk om de bandbreedtevereisten van een bepaalde dienst voor



te stellen en deze toe te wijzen aan een *injection point*. In hoofdstuk 4 werd reeds getoond dat om een optimale configuratie van een bepaald aantal *injection points* te vinden, verschillende permutaties iteratief overlopen moeten worden. Dit optimalisatie probleem werd verder niet behandeld.

In hoofdstuk 5 werden op basis van de uitvoer van de tool, het *daily pattern*, *monthly pattern* en ook heatmaps gemaakt. Het *dialy* en *monthly pattern* werd voor FIA\_RES en FIA\_PRO gemaakt en vergeleken. Voor FIA\_RES liggen de peak-times tussen 21:00 en 22:00 uur, bij FIA\_PRO zijn de peak-times iets minder duidelijk. De meeste peak-times liggen echter wel tussen 9:15 en 12:30 uur. Verder werd er ook nog aangetoond dat voor zowel FIA\_RES en FIA\_PRO er verschillen zijn per SIMBA-zone. De zones twee, vier en vijf hebben een kleine variatie en leunen dicht bij elkaar aan. Terwijl Z1 altijd minder verbruikt en Z3 altijd het meeste verbruikt. Het verschil tussen Z3 en Z1 voor FIA\_RES is  $\approx 190$  Gbps, bij FIA\_PRO is dit  $\approx 4$  Gbps.

Bij het *daily pattern* werden er duidelijke trends waargenomen voor zowel FIA\_RES als FIA\_PRO. Voor FIA\_RES is er een duidelijk verschil tussen de weekdays zonder woensdag, woensdagen en weekends. Bij FIA\_PRO is er enkel een verschil tussen weekdays en weekends. Verder werd er op basis van de klantenaantallen ook nog de contributie per klant vergeleken voor ISAM\_HOSTS, hierbij was te zien dat de FIA\_PRO klanten meer verbruiken per klant dan de FIA\_RES klanten.

Tenslotte werden er op basis van het dialy pattern voor FIA\_RES, heatmaps gemaakt. Deze heatmaps stellen Proximus in staat om de evolutie van trafiek doorheen de dag visueel weer te geven.

Hoewel de analyses uit hoofdstuk 5, in deze masterproef, enkel gemaakt werden voor FIA\_RES en FIA\_PRO is het mogelijk om deze analoog te maken voor andere diensten. Deze analyses en methodologie rond trafiekmatrices bieden waardevolle inzichten over het Proximus netwerk. Deze inzichten helpen Proximus verder bij hun upgradeplanning.

## 6.1 Toekomstige onderzoeksperspectieven

In deze sectie worden enkele toekomstige onderzoeksgebieden en uitbreidingsmogelijkheden gespecificeerd. Het onderzoeksproject zal immers na deze masterproef verder lopen. Eerst worden enkel suggesties gedaan om de tool te verbeteren. Daarna wordt even stil gestaan

bij het zoeken naar een (sub)optimale oplossing voor het simuleren van *injection points* met trafiekmatrices. Ten slotte wordt er iets meer verteld over het voorspellen van de bandbreedtevereisten.

### 6.1.1 Uitbreidingen aan de tool

Er zijn verschillende mogelijk uitbreidingen voor de tool. De interactie met de gebruiker verloopt momenteel volledig via de IDE. Een grafische gebruikersomgeving zou een mooie toevoeging zijn. Hierbij zou de gebruiker de inputbestanden kunnen toevoegen en vervolgens het type van de inputbestanden selecteren. Andere parameters zouden ook allemaal ingesteld kunnen worden zoals het aggregatieniveau, schaal, down of up, .... Op deze manier zou de tool toegankelijker zijn voor eindgebruikers.

Hoewel er tijdens de ontwikkeling rekening gehouden is met performantie van de tool, zou het toch nog nuttig kunnen zijn om een uitgebreide performantie analyse te doen. Hoe sneller de tool werkt, hoe sneller de analyses gemaakt kunnen worden.

De input validatie is ook nog een uitbreidingspunt. In sectie 2.4 werden reeds enkele problemen met de data aangehaald. Sommige van deze problemen zouden automatisch opgelost kunnen worden. Wanneer er metingen zijn voor meerdere dagen in één bestand dan kunnen deze lijnen in principe verwijderd worden. Onvolledige lijnen worden momenteel genegeerd. Deze lijnen zouden ook a priori verwijderd kunnen worden. De meer structurele problemen zoals het ontbreken van informatie, zullen door Proximus opgelost moeten worden.

### 6.1.2 Suggesties tot het verbeteren van de inputdata

Niet alle velden uit de inputbestanden worden gebruikt. Het zou dus nuttig kunnen zijn om deze velden te verwijderen. Een beter validatie aan de kant van Proximus zodanig dat enkele van de scenario's beschreven in sectie 2.4 niet kunnen voorvallen, zou uiteraard ook een stap voorwaarts zijn.

Het gebruik van *multi-valued fields* zou moeten vermeden worden. Een voorbeeld hiervan is ap-ena02als01-h02als00009, hierbij worden drie waarden in één kolom gezet. Het zou veel aangenamer zijn mochten deze velden opgesplitst worden. Deze waarden kunnen vervolgens nog eens opgesplitst worden want zowel ena02als01 en h02als00009 zijn zelf ook *multi-valued fields*. De structuur van de inputbestanden zou er op deze manier volledig

anders uit zien. Het voordeel is dat er tijdens het parsen geen string bewerkingen moeten gebeuren.

### 6.1.3 Optimale oplossing voor trafiekmatrices

In hoofdstuk 4 werd niet verder ingegaan op het zoeken naar de optimale oplossing voor het Proximus netwerk. Het zoeken naar een (sub)optimale oplossing voor dit probleem is namelijk een optimalisatieprobleem dat buiten de scope van deze masterproef valt. Er is dus extra onderzoek nodig naar een geschikt optimalisatie algoritme voor dit probleem. Dergelijke optimalisatiealgoritmen kunnen rekening houden met enkele van de criteria aangehaald in hoofdstuk 4, zoals de hop-afstand of de totale link belasting. Deze criteria kunnen dienen om een waardeoordeel over een bepaalde trafiekmatrix uit te spreken.

Een eerste naïeve aanpak zou kunnen zijn om nodes een random *injection point* toe te wijzen uit de lijst van beschikbare *injection points*. Op deze manier kan een ‘random’ trafiekmatrix opgebouwd worden. Vervolgens krijgt deze oplossing een score naargelang hoe goed ze is. Hoe goed een oplossing is, wordt gedefinieerd aan de hand van een objectieve functie. Het is deze functie die rekening kan houden met bovengenoemde criteria. Om deze score te kunnen berekenen zal het netwerk aan de hand van de trafiekmatrix gesimuleerd moeten worden. De oplossing met de hoogste score wordt bijgehouden. Dit proces wordt dan x keer herhaald ofwel tot een goeie score ofwel tot een maximaal aantal iteraties bereikt wordt.

### 6.1.4 Trafiek voorspellen

Hoewel het nu reeds mogelijk is om bepaalde verwachtingswaarden uit de grafieken te halen die besproken werden in hoofdstuk 5, is het niet mogelijk om voorspelling op lange termijn te maken. Er is dus verder onderzoek nodig naar het voorspellen van trafiekmatrices. Het is ook aan te raden voor Proximus om de tool te gebruiken om historisch data op te bouwen. Dergelijke data kan dan gebruikt worden door toekomstige voorspellingsmodellen. Deze data zal zeker nuttig zijn als er een vorm van *machine learning* zoals neurale netwerken gebruikt zal worden. Deze voorspellingsmodellen zullen dus ook rekening moeten houden met de verschillende trends die besproken werden.

## Bibliografie

- [1] Egemen K. Çetinkaya, Dan Broyles, Amit Dandekar, Sripriya Srinivasan, and James P G Sterbenz. Modelling communication network challenges for Future Internet resilience, survivability, and disruption tolerance: A simulation-based approach. *Telecommunication Systems*, 52(2), 2013.
- [2] P. Cholda, A. Mykkeltveit, B.E. Helvik, O.J. Wittner, and A. Jajszczyk. A survey of resilience differentiation frameworks in communication networks. *IEEE Communications Surveys & Tutorials*, 9(4), 2007.
- [3] Robert D Doverspike, K K Ramakrishnan, and Chris Chase. Structural Overview of ISP Networks. In *Guide to Reliable Internet Services and Applications*. 2010.
- [4] Sandvine Incorporated ULC. 2015 - Global Internet Phenomena Asia-Pacific & Europe. 2015.
- [5] ITU. IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond, M Series, Recommendation ITU-R M.2083-0 (09/2015). 2015.
- [6] Hyojun Kim, Nitin Agrawal, and Cristian Ungureanu. Revisiting Storage for Smartphones. *ACM Transactions on Storage*, 8(4), 2012.
- [7] Tom Lehman, Xi Yang, Nasir Ghani, Feng Gu, Chin Guok, Inder Monga, and Brian Tierney. Multilayer networks: An architecture framework. *IEEE Communications Magazine*, 49(5), 2011.
- [8] Jose-juan Pedreno Manresa and Pablo Pavon-marino. Joint Fault Tolerant and Latency-Aware Design of Multilayer Optical Networks. (Ondm), 2016.
- [9] Eugene S Myakotnykh, Bjarne E Helvik, Jon Kåre Hellan, Olav Kvittem, Trond Skjesol, Otto J Wittner, and Arne Øslebø. Analyzing Causes of Failures in the Global

- Research Network Using Active Measurements Centre for Quantifiable Quality of Service in Communication Systems (Q2S).
- [10] Rysavy Research. Mobile Broadband Explosion. Technical report, 2013.
- [11] SaiShankar N, Debashis Dash, Hassan El Madi and Guru Gopalakrishnan. WiGig and IEEE 802.11ad For Multi-Gigabyte-Per-Second WPAN and WLAN. *ZTE Communications*, 2012.
- [12] Sandvine. 2015 - Global Internet Phenomena Latin America & North America, 2015.
- [13] David Tipper. Resilient Network Design: Challenges and Future Directions. *Telecommun. Syst.*, 56(1), 2014.
- [14] Paul Tune and Matthew Roughan. Internet Traffic Matrices : A Primer. In *Recent Advances in Networking*. 2013.
- [15] Jean-Philippe Vasseur, Mario Pickavet, and Piet Demeester. *Network Recovery*. 2004.
- [16] Google Fiber - Internet speeds to expect - Fiber Help. <https://support.google.com/fiber/answer/6250056?hl=en>, Geraadpleegd op 26/02/2017.
- [17] Cisco Visual Networking Index: Forecast and Methodology Cisco Visual Networking Index: Cisco Visual Networking Index: Forecast and Methodology. *Forecast and Methodology*, pages 2015–2020, 2015.
- [18] Netflix - Aanbevolen snelheid van internetverbinding. <https://help.netflix.com/nl/node/306>, Geraadpleegd op 26/02/2017.

# Bijlage A

## Permutaties

Tabel A.1: Permutaties voor twee injection points op basis van het netwerk in figuur 4.1

Dienst 0	Node D	Node E	Node F	Node G
ip1	25	15	10	5
ip2	0	0	0	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	15	10	5
ip2	25	0	0	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	0	10	5
ip2	0	15	0	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	0	10	5
ip2	25	15	0	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	15	0	5
ip2	0	0	10	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	15	0	5
ip2	25	0	10	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	0	0	5
ip2	0	15	10	0

Tabel A.1: Permutaties voor twee injection points op basis van het netwerk in figuur 4.1

Dienst 0	Node D	Node E	Node F	Node G
ip1	0	0	0	5
ip2	25	15	10	0
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	15	10	0
ip2	0	0	0	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	15	10	0
ip2	25	0	0	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	0	10	0
ip2	0	15	0	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	0	10	0
ip2	25	15	0	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	15	0	0
ip2	0	0	10	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	15	0	0
ip2	25	0	10	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	25	0	0	0
ip2	0	15	10	5
Dienst 0	Node D	Node E	Node F	Node G
ip1	0	0	0	0
ip2	25	15	10	5





# Extraheren en voorspellen van trafiekmatrices als input voor planning van telecommunicatienetwerken

Axl François

Promotoren: prof. dr. ir. Sofie Verbrugge, prof. dr. ir. Didier Colle

Begeleiders: Jonathan Spruytte, mevr. Elise Vivier (Proximus)

Masterproef ingediend tot het behalen van de academische graad van  
Master of Science in de industriële wetenschappen: informatica

Vakgroep Informatietechnologie  
Voorzitter: prof. dr. ir. Bart Dhoedt  
Faculteit Ingenieurswetenschappen en Architectuur  
Academiejaar 2016-2017

