University of Padova

Department of Information Engineering

Master degree in Telecommunication Engineering

Traffic flow optimization for urban xDSL based access networks

Author Enrico Lovisotto

Internal supervisor Prof. Andrea Zanella

External supervisors Prof. Petri Mähönen

Dr. Ljiljana Simić

April 8, 2019

Academic year 2018-2019

Abstract

Internet is currently growing in an unprecedented way, both in terms of users and offered traffic. Software-Defined Networks (SDN) will be necessary to ensure a suitable Quality of Service for users in a more and more complex future, as legacy IP networks are not up to the task because of their configuration inflexibility. Such programmable networks have proven successful in many context, from data-centers to backbone management, but currently no research has analyzed what happens in small-scale scenarios, such as common city access networks. In this thesis we try to quantify how much improvement an SDN approach can give to such a simple infrastructure.

As our case of study, we choose the city of Aachen, located in north-west of Germany in the state of North Rhine-Westphalia, site of RWTH university, which this thesis was written in collaboration with. Since no schematics of its access network were publicly available, we infer its topology from population density and building maps, setting an optimal location for each network switch, from the DSLAMs to the backbone router.

After this design phase, we devise a strategy to distribute the common bandwidth among network users in a *fair* way. Each user is therefore assigned an utility function that links its amount of resources with perceived Quality of Experience (QoE). The best synthesis for these individual evaluations is found in literature as the so-called *Nash arbitration scheme*, equilibrium of the cooperative allocation game played among users.This optimal operation point is compared then with *proportional fairness*, the traditional approach for flow control where each user is assigned a bandwidth proportional to its demands.

Simulating network operations, we show that our approach indeed improves user QoE with respect to legacy techniques of a significant extent, especially when demand is high and therefore network management is more challenging for administrators.

Sommario

Internet sta vivendo in questo periodo una crescita senza precedenti, sia in termini di utenti che di traffico offerto complessivo.

Le cosiddette Software-Defined Networks (SDN) saranno necessarie in futuro per garantire le richieste di Quality of Service (QoS) da parte degli utenti, dal momento che le rete IP tradizionali non sono abbastanza flessibili e configurabili per la crescente domanda.

Queste reti programmabili hanno trovato applicazione in molti ambiti, dal *cloud computing* alla gestione di reti backbone, ma al momento nessuno studio le ha messe alla prova in contesti più comuni, come la rete di accesso di una città. Questa tesi si pone quindi come obiettivo di studiare i limiti delle SDN, valutando qual è il loro impatto in contesti più semplici.

Il nostro caso di studio è, nello specifico, la rete di accesso della città di Aquisgrana, situata nel nord-ovest della Germania, nello stato della Nord-Renania Westfalia. Siccome i suoi schematici non sono pubblici, abbiamo ricavato la topologia di questa rete a partire dalla distribuzione degli edifici e della popolazione, e posizionando poi in modo ottimale gli switch sulla superficie cittadina.

Ultimata la fase di design, ci siamo posti come obiettivo quello di allocare le risorse di rete tra i terminali in modo "fair". Ad ogni utente è perciò assegnata una funzione utilità, il cui compito è quello di stabilire la sua Quality of Experience (QoE) a seconda della banda disponibile. La miglior sintesi di queste valutazioni soggettive è il cosiddetto *Nash arbitration scheme*, ovvero il punto di equilibrio, secondo la teoria dei giochi, del problema di allocazione. La strategia proposta è quindi confrontata con quella tradizionale, detta *proportional fairness*, che assegna ad ogni utente una banda proporzionale alle sue richieste.

Dopo aver simulato la rete, proveremo che il nostro approccio migliora la qualità del servizio per gli utenti anche in questo semplice contesto. L'effetto è rilevante specialmente quando il traffico offerto aumenta, e quindi la gestione della rete da parte dell'amministratore si complica.

Contents

Listing of figures

Listing of tables

1 Introduction

Since the creation of the public Internet, the configuration of the network layer of the protocol stack has been static. Each route has either been discovered by the device itself or set by the hand of the administrator: in both cases no tools were available to quickly modify the behaviour according to changes in traffic demand or network structure.

In last years, this inflexible paradigm was questioned, as the complexity and sheer amount of data exchanged in the Internet grew dramatically, as shown in Fig. 1.1. Moreover, traffic is not yet expected to stabilize, according to CISCO predictions, plotted in Fig. 1.2 [3].

In order to prevent the "ossification of the Internet" [4] and adapt to these new scenarios, it is in fact essential to separate the *data plane*, the actual physical devices forwarding packets, from the *control plane*, responsible for deciding routes.

The latter operations are supervised by a logically centralized *controller*, whose task is to orchestrate involved routers and switches based on current and forecast network status. This moves the routing logic from the single component to a fully programmable core entity.

In case of congestion of a particular link, for example, a data flow can be diverted to another path, possibly increasing Quality of Service (QoS) for all users. This can be very interesting in cable networks, where gridlock is indeed the main issue for administrators.

Given the demand for connectivity and bandwidth, Software Defined Networks (SDNs) are now a ground-breaking approach to further improve the performance using unconventional approaches. Not only flow control, but also user mobility prediction could be exploited by the programmable controller logic.

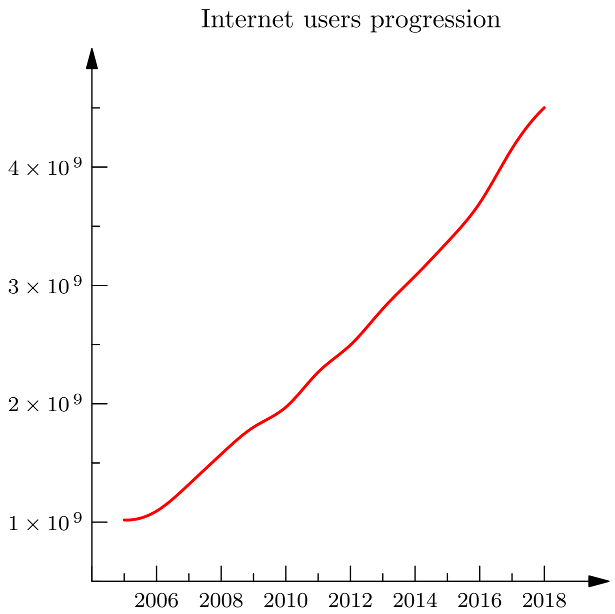


Figure 1.1: Number of Internet users has steadily increased for the past fifteen years.

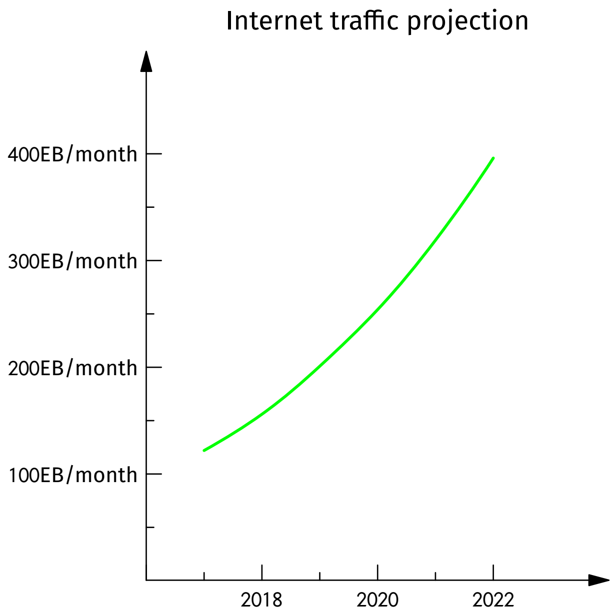


Figure 1.2: Internet traffic is expected to explode in the next years.

However, as every tool, its makings have to be coupled with a robust knowledge on its limits. This information is therefore crucial to make the most out of SDN as, from a methodological point of view, alleged once-and-for-all solutions often fall short when context changes from the expected one: .

As mentioned before, SDNs are indeed popular nowadays when dealing with very high bandwidth link management in scenarios that are complex because of either topology or applications demand [5].

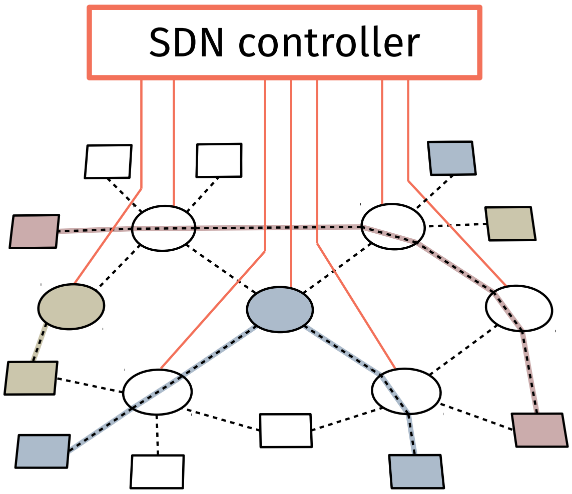


Figure 1.3: Data flow of users (squares) are tuned and managed by the SDN controller via the routers (circles).

In this thesis we aim then to understand what happens at the other side of the spectrum, i.e. when dealing with static and simple access networks. At the moment no study has been devoted to understand and quantify how much a controller can improve QoS in these simple, but very common, settings.

We specifically focus on the access network of Aachen, historical city in the German state of North Rein-Westphalia. As the actual schematics are not available to the general public, our network topology is inferred by residential buildings and population distribution across the area, solving an optimization problem.

After this design step we simulate the normal operation of the infrastructure, where a certain number of clients requires traffic for their applications. In this setting we try to assess whether an SDN inspired approach can effectively increment Quality of Experience (QoE) for the users.

While state-of-the-art utility functions are employed as service-specific quality measures, a comprehensive assessment is performed using a game-theoretical framework, known as *Nash arbitration scheme*. Obtained operation point is the only one that guarantees a *fair* allocation of resources among users and it is proven to maximize the product of the utilities.

More specifically then, our goal is to compare traditional network management technique to this novel approach in a real-world scenario.

This thesis is structured as follows.

First, an overview of past works related to this thesis scope is provided in chapter 2.

Then, chapter 3 introduces the theoretical instruments employed in this analysis. In section 3.1, we will show how relevant information from Aachen building and road map can be extracted in order to design a proper access network based on city topology and population density, in section 3.2. section 3.3 will detail how this infrastructure can be tuned for maximize user perceived QoE.

After applying these procedures to our case of study, final results are collected and commented in chapter 4, split again into geographical analysis, network design and optimization, in section 4.1, section 4.2 and section 4.3 respectively.

Final remarks and considerations are eventually discussed in chapter 5.

2

State of the art

As mentioned in the introduction, the first step of this thesis is the estimation of the Aachen city network, given publicly available information on city topology and population density.

Once extracted the relevant feature of the geographical area, the designing of such a network is a matter of connecting all terminals to a central unit, aggregating user traffic to a single end-point.
This is known in literature as the Steiner tree problem. On a graph $G = (V, E)$ a set $T \subset V$ of nodes have to be reached from a root $r \in V$: the goal is to select the subset of edges in E that activates those links at minimum cost.
This problem was proven to be NP-hard and has been extensively studied and solved both with exact and approximated solutions [6, 7, 8, 9].
A reference formulation of it is given in (2.1), where active edges $e \in E$ are marked by a binary variable x_e and a cost c_e. Function $\delta^+(\cdot)$ and $\delta^-(\cdot)$, instead, give the edges entering or exiting their argument, respectively.

$$\begin{aligned} \max \; & \sum_{e \in E} x_e c_e \\ \text{given} \; & \sum_{e \in \delta^-(j)} x_e \begin{cases} = 1 & j \in T \\ = 0 & j = r \\ \leq 1 & j \in V \setminus (T \cup \{r\}) \end{cases} \\ & \sum_{e \in \delta^+(S)} x_e \geq \sum_{e \in \delta^-(t)} x_e \quad \forall S \subset V, \text{ with } r \in S, \; t \in V \setminus S \\ & x_e \in \{0, 1\} \; \forall e \in E \end{aligned} \tag{2.1}$$

In our case, though, reachability is not sufficient to declare a user connected, as its traffic has to be supported by all links in the path to the root as well.

Accounting for this factor, Steiner tree was extended adding flow conditions for bandwidth and link types of different capacity by Andrews *et al.* [1]. Moreover, here fixed edge activation cost c_e becomes proportional to the bandwidth used and a fixed cost for the activation is considered, resulting in the scenario of Fig. 2.1.

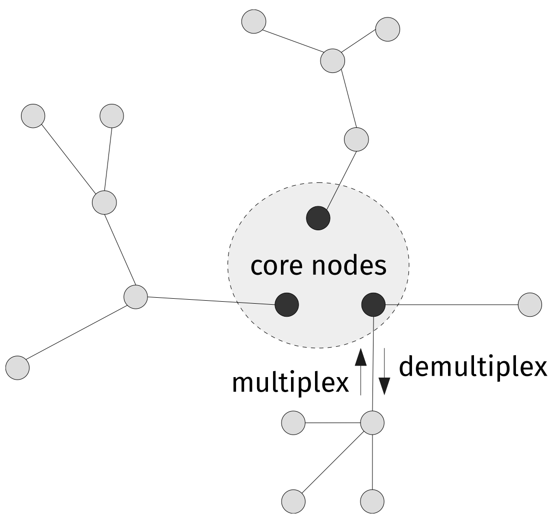


Figure 2.1: Each intermediate node in the tree merges the flows coming from the leaves and splits the return ones [1].

A mathematical formulation describing the optimal solution is given in their paper, and it is coupled with an approximated solver algorithm obtained via linear programming relaxations, whose penalty with respect to the optimum is bounded.

This approach, though, does not take into account the technological constraints, such as a maximum number of ports in traffic aggregators and limits in cable length.

This last point is addressed in Mitcsenkov *et al.* [10], where they consider for the same setting various technologies, such as Passive Optical Networks (PONs), Active Ethernet and VDSL, each one with its own constraints.

Moreover, the access network is built not from an abstract model, but from a set of reference topographic scenarios, such as suburban, town or city, in increasing population density.

Here the mathematical problem, with exact solution but infeasible computation, is coupled with heuristic algorithms tailored to the given physical layer assumptions. Both are run on reference maps, ranging from 0.5km^2 to 4.7km^2 of surface and from 400 to 20.000 number of users. These trials show the viability of the approach, as the approximated solution is within 10% to 15% the theoretical bound provided via an ILP solver, while being still fast to compute.

These studies are useful insights on how to design a practical access network. Proposed tools are not enough, though, as either the conditions or the scale of our reference scenario are out of reach for them: as an example, Aachen has an estimated number of 40.000 subscribers spread across 160.85km^2, both unfeasible for state-of-the-art algorithms.

Therefore, new approaches will be devised in this thesis, in order to properly solve the problem and possibly get the approximation closer to the theoretical bounds.

Once built, the resulting network has to be properly managed in order to offer a decent QoS to end users, for example adopting congestion avoidance and load balancing strategies. However, enforcing these policies in traditional IP networks is a daunting task for administrators [11].

This is mainly caused by the need to use vendor-specific interfaces to specify switching units behavior, most notably setting and updating routing tables and requesting network status report. This inhibits a global orchestration across the whole infrastructure, as specific quirks and capabilities of each device have to be carefully abstracted. This operation needs to be repeated every time one of the protocol changes or a new kind of items is added. Therefore, automatic reconfiguration upon failure or dramatic load change is virtually non-existent in such systems.

Moreover, IP networks are now mostly *vertically integrated*, meaning that policy specification and implementation, known as *control plane* and *data plane* respectively, are allocated in the switching units. This reduces system flexibility and resilience as, for example, a simple change of routing protocol needs careful planning and a lot of man hours to be put in practice.

SDN is an emerging paradigm that tries to seize the problem by the roots, finally separating data and control plane. While the former remains in the devices, the latter logic is implemented in controller. Either as a physical device itself or a distributed entity, it can be programmed to put in practice high-level policies without the administrator to manually delve into all the details.

In order for the controller to operate and manage all diverse switching units, a proper vendor-agnostic communication interface has to be designed. The first widespread example of such a protocol was OpenFlow, proposed in 2008 by the Stanford computer science department [12].

Each OpenFlow device has multiple tables to match incoming packets to a programmable set of rules: these allow modification, forwarding and dropping of data, along with sending reports of anomalies and statistics to the controller. According to installed policies, each network unit can then act as a switch, a router, a firewall and this behaviour can be easily modified by the administrator.

The new possibilities and the practical viability of the protocol started to attract not only academical contributions, for instance NOX Network Operating System (NOS) [13], but also industrial interested from many preeminent actors. Google [14], WMware [15] and many operators around the world [16] started to experiment and deploy software-defined solutions in their infrastructure, reporting performance and flexibility gains.

Given our reference scenario, the levers offered by an SDN approach are limited. As mentioned before, our reference topology is a rigid and hierarchical tree: while this minimizes infrastructure cost, no routing is possible in this context.

The only tool left on the table is then *flow control*, which is the ability to give or revoke priority from stream of packets. Traditionally this technique has been employed for congestion avoidance, primary example being distributed Transmission Control Protocol (TCP) strategies to match the bitrate with the actual link capacity [17]. Nowadays, it is becoming necessary in order to assign users the proper amount resources their application requires. For instance, video streaming shows a different profile than traditional web browsing and this factor needs to be taken into account when administrating the network.

In order to perform a decent and user-tailored resource distribution, the link between actual connection metrics and user perceived quality of the service is needed.

Many research groups estimate QoE under a wide spectrum of network conditions using the so-called Mean Opinion Score (MOS), the average subjective evaluation among a group of candidates. Georgopoulos *et al.*, Laghari *et al.* focus on the actual measurement and try to fit what found with a shifted power function of the bandwidth [18, 19]. Instead, Reichl *et al.* go for a different approach, as they *a-priori* suppose QoE to be a logarithmic function of available capacity: this hypothesis is taken in analogy to the Weber-Fechner Law [20].

This law, formulated by Gustav Theodor Fechner and its student Ernst Heinrich Weber, is a key principle in psycho-physics that connects the magnitude of a *stimulus* to its perceived intensity.

$$k = \frac{\mathrm{d}S}{S} \tag{2.2}$$

$$dp = k \frac{dS}{S} \implies p = k \log \frac{S}{S_0} \tag{2.3}$$

where p is the extent of the perception given by the external stimulus S. k and S_0 are instead constants and depend on the specific physical quantities considered.

Former equation (2.2), known as *Weber's contrast*, This was supposed after performing an experiment, consisting in a man judging the weight of an object: it was observed that constant increments in the weight were more noticeable when the object itself was light. Thus it was concluded that the differential stimulus is inversely proportional to the stimulus itself.

Latter equation (2.3), tries to move forward this preliminary result as general increments are allowed. The final relation between stimulus and perception is therefore logarithmic.

These investigations give us analytical tools to effectively simulate and reproduce satisfaction for each user.

Once individual metrics are computed, it is necessary to aggregate all these QoEs into a global score, in order to choose the best among all possible network configurations.

A game theoretical framework is proposed in literature in order to tackle this problem. When the network is operational, in fact, there is an implicit competition between each user wanting to maximize its own *utility* at the expense of others bandwidth. The main task of the administrator is then to guide individual demands to a stable and *fair* operation point.

"Fairness" is, from a mathematical point of view, an elusive concept, but it can be seen here as a condition where all users enjoy the service without being overly penalized.

Pareto-optimal strategies are suggested by Douligeris *et al.* in order to capture this requirements, as no one can increment its own utility without worsening the experience of at least one of its fellows [21]. No criterion is however provided in order to choose among the possibly infinite number of these solutions.

Mazumdar *et al.* eventually propose the so-called *Nash arbitration scheme*, a stable operation point of the cooperative game played among users, located in throughput space [22]. Such point is proved to exist and be the only one satisfying symmetry, independence of irrelevant alternatives and Pareto optimality. Moreover, they show that this strategy maximizes the product of individual user utilities, giving a straightforward way to compute it in practice.

It is important to differentiate between Nash arbitration scheme from the popular Nash equilibrium, as the former arises from a *cooperative* game, while the latter from a *competitive* one. Such equilibria are situations where no user can increment its own utility if others behaviour stays the same [23]. These points are typically Pareto sub-optimal and therefore not relevant for our purposes.

An example of this concept is the *prisoner's dilemma*, a non-cooperative game played between two actors A and B. These two individuals are suspected of a crime, but the prosecutor lacks evidence for this, while he has for some minor charges. Both of the criminals, who have no way of communicating to each other, can either confess the main offence or stay silent.

	A confesses	A stays silent
B confesses	-2, -2	0, -3
B stays silent	-3, 0	-1, -1

Table 2.1: Each couple of strategies leads to two utility scores (years in prison), the first for B and the second for A.

Judging from Table 2.1, the best course of action, meaning the Pareto optimal strategy, is for the two of them to stay silent. This point, however, is not a Nash equilibrium, because each one has incentive to betray the other and spend no time in prison.

This toy example shows how Nash equilibria, originating from competitive games, are often not optimal with respect to the global utility. In our context, then, the operator has to enforce a fruitful coordination, the Nash arbitration scheme, among players carefully tuning their packet flows.

3 Methodology

This chapter will present which theoretical tools and frameworks are employed in our analysis.

First, in section 3.1, we show how relevant information about city topology is extracted from publicly available data, namely population density, streets and buildings position.

Then in section 3.2, we will show how to design a network given valid paths across the city area and supposed user demand. As mentioned earlier this will be performed solving an optimization problem, akin to Steiner tree.

Finally, obtained network is optimized in section 3.3 with respect to user requirements and applications, distributing available network resources in a *fair* way. Global user utility will be maximized using a game-theoretical framework, called *Nash arbitration scheme*.

3.1 Geographical analysis

The city of Aachen is located in the north-west of Germany, in state of North Rhine-Westphalia. Its district has a surface of 160.85km^2 and a population of 244,951 citizens.

Although medium sized, the city is an important telecommunication node between Germany and the neighbour countries of Belgium and Netherlands. The LambdaNet backbone, owned by *euNetworks Managed Services GmbH*, crosses in fact the city and provides direct connection to public Internet. Its map, built by “The Internet Topology Zoo” project [24], has been plotted in Fig. 3.1.

In this thesis we will then suppose that the access network connects all Aachen buildings to this main backbone via a single Point of Presence (PoP), located in the industrial district of the city.

Unfortunately, schematics for such network are not publicly available, so we have to perform what it is called an *educated guess*, meaning a good estimation based on available information.

The evaluation will be performed using OpenStreetMap [25] in conjunction with the *Open Data Portal* of the city of Aachen: * the former provides buildings and roads positions, while the latter describes how population is distributed across the city districts.

All this information can be visualized in the map of Fig. 4.1, in the section 4.1.

Due to the level of detail of these datasets, two assumptions are needed to proceed and extract a reasonable diagram for the access city network.

First, we suppose cables to be put along streets and not to cross (even public) terrains. This is common practice, since roadworks are usually exploited to perform maintenance and build new parts of the communication network.

Second, we consider the population of a given area to be uniformly distributed across a fraction of its buildings, so-called *residential* ones, randomly picked among all the constructions.
We have to take this strong hypothesis because the OpenStreetMap dataset lacks information about the building use and height in most entries.

These two points can be accepted in this work as the end goal is to study how the access network of a city like Aachen behaves, not to replicate it in perfect detail.

*Please refer to `http://daten.aachen.de` for further information and licensing.

Map of LambdaNet

Figure 3.1: LambdaNet is a national backbone that serves all major German cities and connects the country to the rest of Europe.

3.2 Network design

This information is then condensed in an abstract graph $G = (V, E)$, with streets as edges and road crossings as vertices. The former were given corresponding lane length, while the latter were assigned the supposed number of people living in the surrounding area.

More specifically, each node $i \in V$ is assigned a number of users u_i to serve and, since that they represent a physical line, edges in E are given a length value l_e: both these parameters will be used later to evaluate the access network cost.

In this chapter we will exploit this information to find the optimal network configuration, given some assumptions and requirements derived from best practices in access network design [26].

3.2.1 Topology considerations

As depicted in Fig. 3.2 we suppose our access network to be made of layer-2 type switches and to be logically shaped as a tree. This is indeed common practice in such access networks, where more complex and elaborate topologies are too expensive and offer no substantial benefit [26].

In this configuration the path from users to the provider mainframe is fixed and must cross two kinds of intermediate nodes, a Digital Subscriber Line Access Multiplexers (DSLAMs) and a router.

From a technological point of view the network is considered to be relatively modern, since the infrastructure has been renewed on the past years in conjuction with works on main city roads.

That is the reason why we suppose all main links to be fiber optic running state-of-the-art VDSL/VDSL2. The minor fraction of legacy ADSL and copper-cable users can be well approximated as VDSL connections at the same distance, in terms of bandwidth and other network metrics.

In order to guarantee a suitable QoS, all connected network components have to be close enough to each other: this is taken into account though a maximum distance parameter d_M.

Finally, each switch is allowed to serve a limited number n_M of lower level nodes, given by the number of physical ports of the device.

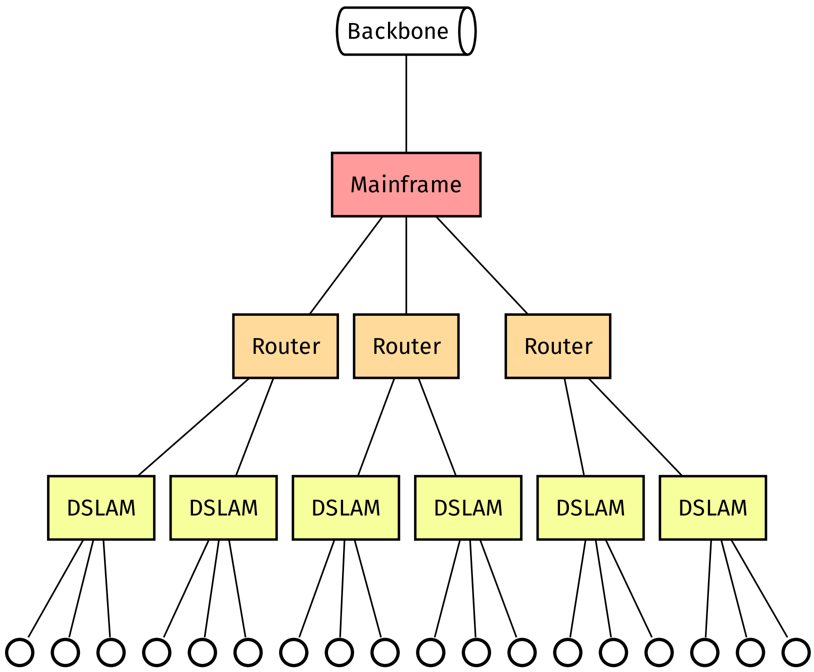


Figure 3.2: A layered tree access network connects users (circles) to the Internet backbone

3.2.2 Solution approach

In smaller contexts, a manually design of the network suffices to meet all the technological constraints while being reasonably cheap. This is not our case, since the set of possible topologies is far too vast for a manual evaluation: a programmatic strategy is then necessary to proceed.

Problems on graphs similar to the one we face are often solved using either ILP or an heuristic approach [8, 7, 27, 9].
The former is a powerful mathematical tool that finds the best possible solution to the problem, but it is very demanding with respect to computational resources and time.
The latter instead does not strive to give the optimum, but can hopefully achieve decent results in a more reasonable amount of time.

A mathematical model can be written to describe the multi-layered system as a whole, but its complexity would have made it impossible to handle by any solver, both in terms of number of variables and constraints.

To overcome this issue a different way of designing the topology has to be devised. Instead of positioning all the nodes at once, the proposed algorithm would place the leaves of the tree, meaning the DSLAMs, first and then move up to the higher-level elements.
This is closer to what is done in practice, as each step is examined and evaluated according to criteria, such as soundness and future-proofing of the infrastructure, that are difficult to explain to the solver.

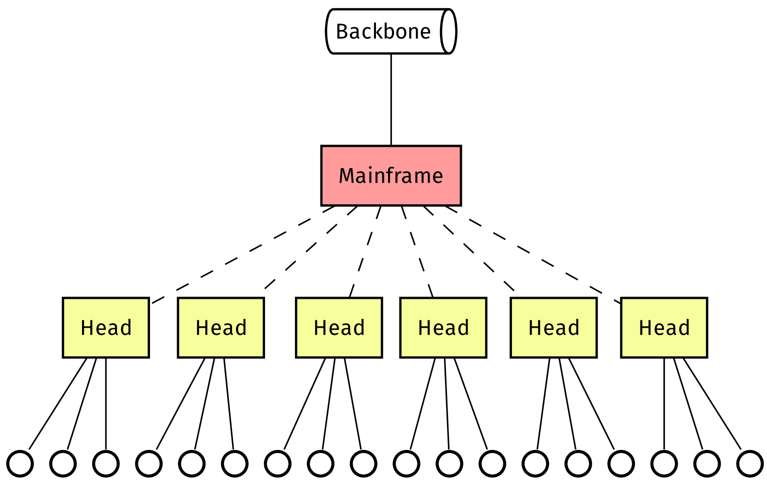


Figure 3.3: Each *head* aggregates the traffic of all nodes in its *cluster*.

The network topology moves then from the one in Fig. 3.2 to the simplified setting of Fig. 3.3.

As apparent in the diagram the solver must now take into consideration the cost of the nodes that have been omitted from the tree. This is accounted as a lump sum for the connection of each network switch, called from now on cluster *head*, to the mainframe both in terms of cables and intermediate nodes.

Both the exact and approximate approach that will be proposed in this thesis will build the access network in this fashion, starting from the periphery and moving towards the core of the network.

All relevant parameters have been collected in Table 3.1 and will be taken for granted from now on.

Variable	Description
$G = (V, E)$	Graph describing the city topology
$T \subseteq V$	Set of terminal nodes
$l_e = l_{ij}$	Length of edge $e = (i, j) \in E$
u_i	Number of users at terminal $i \in T$
d_M	Maximum distance from a terminal and its root
n_M	Maximum number of terminals per tree
c_r	Cost of a single subtree root node, plus mainframe connection
c_f	Cost of a fiber optic cable per meter
c_e	Cost of roadwork excavation per meter

Table 3.1: Problem parameters, divided in topology specific ones, technological limits and costs.

3.2.3 ILP FORMULATION

In order to express the optimization problem in a convenient way, we arrange our data as follows.

A direct graph $G' = (V \cup \{r\}, A)$ is induced on top of the G, where the set of arcs A is defined as follows.

$$A = \{(i, j), (j, i) \ \ \forall\{i, j\} \in E\} \cup \{(r, j) \ \forall j \in V\} \tag{3.1}$$

In (3.1) each undirected edge in E is doubled with the two corresponding directed arcs; then an artificial node r is added to the vertices set and connected to each of the nodes in V.

Each arc $(i, j) \in A$ is assigned a length l_{ij}, in meters, given by the geographical distance between its endpoints. Artificial arcs (r, j) do not correspond to physical connections and so $l_{rj} = 0 \ \ \forall j \in V$.

With this setup our network access configuration will simply be a direct tree, or *arborescence*, with root in r, as depicted in Fig. 3.4.

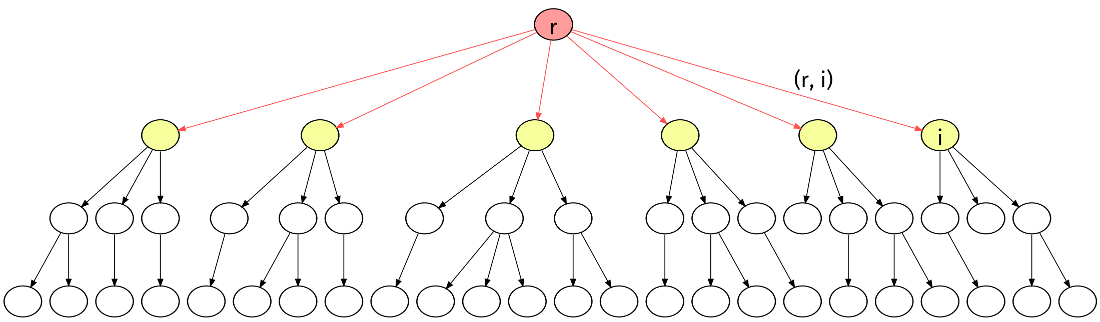


Figure 3.4: In the final solution, additional arcs (r, i) connect artifical node r to all the roots, making the whole structure an arborescence, instead of a forest.

Because of the system requirements we also have to keep track of the distance d_i of each node $i \in V \cup \{r\}$ from its head and the number of users n_e served by each link in A, ensuring they do not exceed their limits.

Given this setup, our optimization problem can be written as follows.

$$\min_{\substack{\{x_e\}_{e\in E}\\ \{u_t\}_{t\in T}}} \left(\sum_{t\in T} d_t\, u_t\right) c_c + \left(\sum_{e\in E} x_e\, l_e\right) c_e + \left(\sum_{e\in\delta^+(r)} x_e\right) c_r \tag{3.2}$$

$$\text{subject to} \quad \sum_{e\in\delta^-(j)} x_e \begin{cases} = 0 & j = r \\ = 1 & j \in T \\ \leq 1 & j \in V \setminus T \end{cases} \tag{3.3}$$

$$\sum_{e\in\delta^+(r)} x_e \geq 1 \tag{3.4}$$

$$\forall j \in V \cup \{r\},\ d_j \leq \left(\sum_{e\in\delta^-(j)} x_e\right) d_M \tag{3.5}$$

$$\forall (i, j) \in A \ \begin{cases} d_j - d_i \geq l_{ij}\, x_{ij} - d_M\,(1 - x_{ij}) \\ d_j - d_i \leq l_{ij}\, x_{ij} + d_M\,(1 - x_{ij}) \end{cases} \tag{3.6}$$

$$\forall e \in A,\ n_e \leq x_e\, n_M \tag{3.7}$$

$$\sum_{e\in\delta^-(j)} n_e - \sum_{e\in\delta^+(j)} n_e = \begin{cases} p_j & j \in T \\ 0 & j \in V \setminus T \end{cases} \tag{3.8}$$

$$\sum_{e\in\delta^+(r)} n_e = \sum_{i\in T} u_i \tag{3.9}$$

$$\forall e \in A,\ x_e \in \{0,\, 1\},\ n_e \in \mathbb{N} \cup \{0\} \tag{3.10}$$

$$\forall j \in V \cup \{r\},\ d_j \geq 0 \tag{3.11}$$

To clear the notation, we have defined functions $\delta^+,\ \delta^- : V \to \mathbb{P}(A)$ associating each node with the out-going and in-going edges respectively.

$$\begin{aligned} \delta^+(j) &= \{(j,\, k) \in A\} \\ \delta^-(j) &= \{(i,\, j) \in A\} \end{aligned} \tag{3.12}$$

The problem is set to minimize the objective function (3.2) that sums up the cost of optical fiber lines, roadworks and the total price of *head* switching units.

The first constraint (3.3) forces the terminals to be connected to our network and sets the number of in-going arcs to be at most one, which is a necessary condition for the network to be a directed tree.

As the leaves are set to be part of the network, r has to be as well by (3.4). It will then be the root node of the resulting tree, as by construction of G' node r has no in-going arcs.

The next equations deal with the variables d_j, distance from the tree root. First, in (3.5) this quantity is limited by d_M if the node is reached by the network, otherwise it is set to zero.
On the other hand (3.6) guarantees the consistency of this metric between two connected nodes, forcing target node distance to be the source one plus the link length.
Implicitly the latter prevents the resulting network to have loops, necessary for our solution to be a proper arborescence.

The last needed metric for limiting the possible solutions is the number of users each link can handle, n_M. This upper limit for n_e is set in (3.7) such that it has to hold only for active edges, and then the count of the users from leaves to each sub-root is performed in (3.8), which has the same form as a flow-conservation clause.
All such flows must converge towards the root r for (3.9): this forces the network to be connected, finally giving it the wanted shape.

Variable domains are eventually specified in (3.10) and (3.11).

Overall, the model requires $|V| + 1 + 4\,|E|$ variables and $3\,|V| + 2\,|T| + 4\,|E| + 1$ constraints, both of which are $O(|V|)$ for sparse graphs like the one we are working on.

3.2.4 Heuristic algorithm

The mathematical problem described in the previous section can be effectively solved only for small instances, i.e. sparse graphs with up to one hundred nodes.
In fact, when tested on our specific case with tens of thousands of nodes and edges, the program could not output the solution within a reasonable amount of time and resources.

An heuristic approach had to be devised: for the peculiarities of the problem it is indeed suitable a *greedy* approach, inspired by hierarchical clustering.

The basic idea is to progressively join single nodes of the graph in bigger and bigger *clusters* until the total cost decreases: once a merge results in a more

expensive network, the algorithm stops.
Such merges are allowed whenever the mentioned QoS constraints are met and adjacent subsets are preferred. To be precise, distance between each couple of groups is defined as the distance of the closest elements: this is done to privilege more cohesive and compact pairs.

This procedure is repeated until all possible choices have been considered or the next merge increases the cost of the network.

Pseudo-code is available in Algorithm 3.1. As defined in Table 3.1, parameters n_M and d_M are the maximum number of nodes per cluster and the maximum distance between vertices in the same cluster, respectively.

Algorithm 3.1 Heuristic solver

```
C = ∅
∀ t ∈ T add singleton {t} to C
mark all couples C_i, C_j ∈ C^2 as mergeable
cost = OBJECTIVE_FUNCTION(C)

stop = False
repeat
    pick C_i and C_j the two closest clusters in C
    d_ij = diameter of cluster C_i ∪ C_j
    n_ij = number of users inside C_i ∪ C_j
    if d_ij < 2 d_M and n_ij < n_M
        C' = {C_1, ..., C_i ∪ C_j, ...}
        current_cost = OBJECTIVE_FUNCTION(C')
        if current_cost > cost
            stop = True
        else
            C = C'
        merge C_i and C_j
    else
        mark the couple C_i and C_j as unmergeable
    if ∄ C_i, C_j ∈ C^2 mergeable
        stop = True
until stop = False
return C
```

The cost of each sub-network is not evaluated on the best possible configuration, but instead goes for a sub-optimal one.
This is required for the algorithm to be feasible, as the Steiner-tree-like prob-

lem that it has to be solved in order to connect all cluster nodes to a common sub-root is yet again too complex.

As it can be seen in Algorithm 3.2, each node close enough to the cluster is evaluated as a candidate root of the corresponding spanning tree. The network is then simply built joining the minimum paths between the best of those and the terminals of the set.

Algorithm 3.2 Approximated objective function

```
function OBJECTIVE_FUNCTION(C)
    total_cost = 0
    for all c ∈ C
        best_cost = +∞
        for all v ∈ V close to c
            T_v = ⋃_{t∈C} minimum path from v to t
            cost_v = cable cost of T_v + excavation cost of T_v
            if cost_v < best_cost
                best_cost = cost_v
        total_cost += best_cost
    return total_cost
```

For now all these approximations are mandatory for the algorithm to be fast enough to deal with our case of study, but in section 4.2 they will prove to be good ones, i.e. to be close to the theoretical optimum.

3.3 Network optimization

Previous optimization steps returned a plausible topology for Aachen city access network.

As mentioned earlier, the next step we take is to optimize it, in order to assess whether the more flexible framework provided by SDN can benefit the overall performance.

3.3.1 Network structure

As presented in section 3.2, the obtained network topology is organized in a hierarchical tree of switches, whose task is to merge all uplink communications towards the mainframe and split the downlink ones among the various destinations.

This structure resembles what was previously described in Fig. 3.2, and is further detailed in Fig. 3.5 and Fig. 3.6.

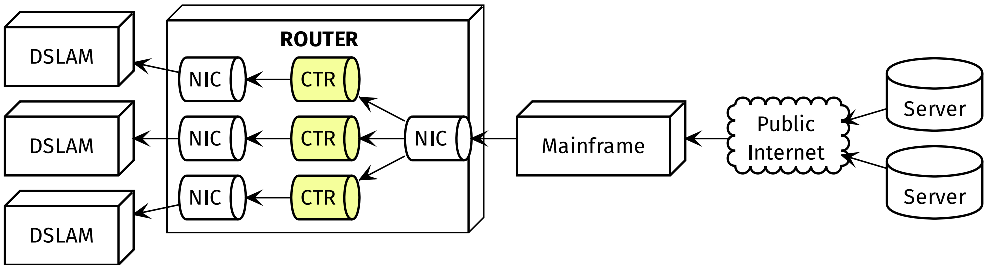


Figure 3.5: Logical node structure for downlink traffic.

As shown in these diagrams, each outgoing flow passes through a controller, whose task is to govern and limit the data rate before it enters the Network Interface Card (NIC). All these units can be coordinated by the central administrator in order to give or revoke priority from a given source.

Since in our setting all data flows are supposed to be Constant Bitrate (CBR), deciding bandwidth allocation is sufficient to provide users the best service.

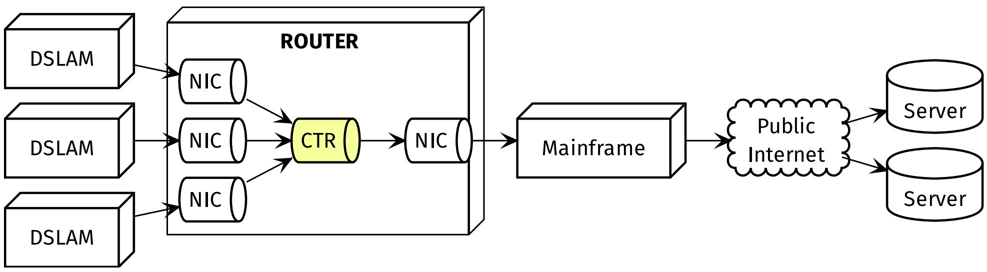


Figure 3.6: Logical node structure for uplink traffic.

3.3.2 Quality of Experience

In order to improve the network, first we have to define what *improvement* means for us. We decide to maximize user QoE, defined as a number in $[0, 1]$, ranging from unusable to perfect link. Each user i is then given a *utility function*, mapping available bandwidth ρ_i to perceived quality.

The functions employed in this thesis were obtained by various research groups collecting user opinion of the Internet service under different network conditions [18, 28]. These studies suggest a precise link between QoE and bandwidth, described by (3.13):

$$u(\rho) = a\,\rho^{b} + 1 \tag{3.13}$$

where a and b are application specific coefficients and ρ is the assigned bandwidth.

More specifically, our network considers two different use-cases: traditional web-browsing and video streaming. Videos are either Low Definition (360p), Medium Definition (720p), or High Definition (1080p). Following *Google Video Quality Report*, each one of those is experienced by 5%, 10% and 85% of the users respectively, as shown in Fig. 3.7 [2].

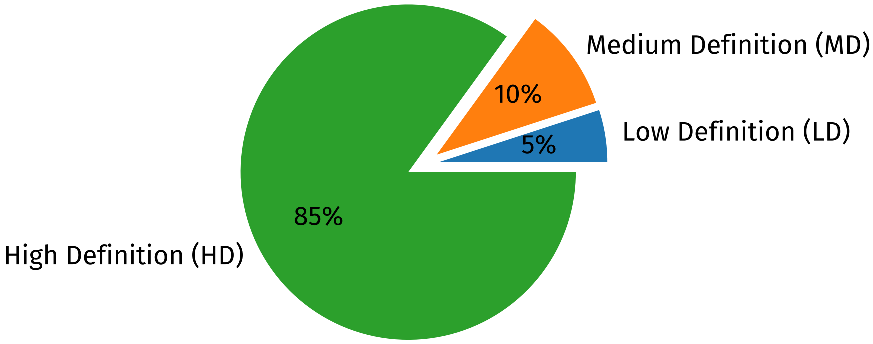


Figure 3.7: Distribution of video quality among video streaming users [2].

Parameters a and b are then tuned in order to properly link user perception of the different services to available resources, as shown in Fig. 3.8.

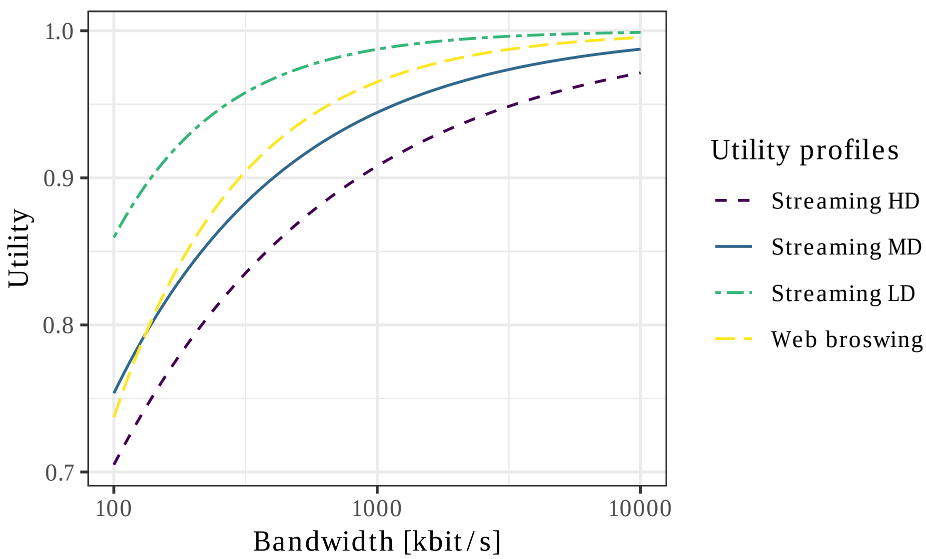


Figure 3.8: Utilities vs available bandwidth for a given service.

Looking at the four trends, it can be seen that High Definition (HD), Medium Definition (MD) and Low Definition (LD) video streaming are in decreasing order of bandwidth demand, as expected. Instead, traditional web browsing puts itself between MD and LD.

3.3.3 Fairness on resource allocation

In order to manage the network optimally, we need to reach an operation point $\vec{\rho}$ in the bandwidth space that balances demands of all parties involved.

Traditionally, the *proportional fairness* principle is applied when handling different flows. According to this rule, each switching unit allocates resources in *proportion* to user request. This has proven to be a reliable way to distribute bandwidth, but it does not take into account the application-dependent service quality perception.

Each user tends to maximize this subjective metric, called *utility*, asking for more and more bandwidth, but doing so it harms the QoS of its fellows. The Internet Service Provider (ISP) has then to act as an arbiter, allocating available resources in a fair way.

The best solution of this problem is proven to be the so-called *Nash arbitration scheme* of the *game* played among these actors [22]. Such unique point satisfies all desirable properties: symmetry, independence of irrelevant alternatives and Pareto-optimality. Moreover, it can be found simply maximizing the product f of all the utilities: this gives us a straightforward criterion to rank all possible network configurations. We then define

$$f(\vec{\rho}) = \prod_i u_i(\rho_i) \tag{3.14}$$

where ρ_i, i-th component of $\vec{\rho}$, is the bandwidth assigned to the i-th user.

In section 4.3 we will then show if the traditional approach to manage an access network is indeed fair or not, with respect to the optimal strategy.

3.3.4 Flow balancing optimization

As anticipated in the previous section, the fair working point is identified as the Nash arbitration scheme, i.e. the vector $\vec{\rho}$, that is solution of the following mathematical problem [22].

$$\arg\max_{\vec{\rho}} \log f(\vec{\rho}) = \arg\max_{\vec{\rho}} \sum_{i=1}^{n} \log u_i(\rho_i) \tag{3.15}$$

$$\text{given} \quad \sum_{i=1}^{n} \rho_i \leq \rho_{MAX} \tag{3.16}$$

$$\forall j \in D, \quad \sum_{i \in DSLAM_j} \rho_i \leq \rho_{D,MAX} \tag{3.17}$$

$$\forall j \in R, \quad \sum_{i \in ROUTER_j} \rho_i \leq \rho_{R,MAX} \tag{3.18}$$

where n is the total number of users and each DSLAM and router, belonging to sets D and R, are assigned a subset of users $DSLAM_j$ and $ROUTER_j$ and a maximum bandwidth $\rho_{D,MAX}$ and $\rho_{R,MAX}$, respectively.

This can be seen as a *water-filling* problem, as a limited resource ρ_{MAX} has to be allocated maximising a concave objective function.

Unfortunately, however, our case is complicated by the two additional constraints (3.17) and (3.18): no algorithm is currently known in literature able to solve the problem optimally [29].

A sub-optimal solution has then to be searched using a heuristic procedure. Given the monotonicity of the objective function, Algorithm 3.3 is employed.

Algorithm 3.3 Flow optimization algorithm

Initialize all users, each with their utility function u_i
Set $\rho_i = 0, \forall i = 0, \ldots, n$
Set stop_condition = False
while stop_condition is False
  Randomly choose k among $\{1, ..., n\}$
  Perturb ρ_k of a uniform random quantity in $[0, K]$
  if ρ_k does not respect constraints (3.17) and (3.18)
    Revert perturbation on k
  if Objective f improvement is negligible in the last L iterations
    stop_condition = True
  Decrement K
return $\vec{\rho}$

A randomly picked bandwidth ρ_i is iteratively incremented each round of a uniform quantity in $[0, K]$: such perturbation remains unless any constraint of the mathematical problem is violated. Once the operation is no more beneficial, i.e. the increment is below a certain tolerance, the algorithm stops.

The rationale behind this is akin to what happens in *simulated annealing*, a heuristic search algorithm where perturbation of constantly decreasing size are applied to the starting point in the hope of converging toward a viable solution [30].

4 Results

This chapter shows what we obtain applying the tools presented in chapter 3 in our specific context, the city of Aachen.

As its theoretical counterpart, here we first extract relevant geographical features from publicly available maps, in section 4.1.

Then, in section 4.2, once the abstract city topology is known, a network is designed on top of it, solving an optimization problem both using an exact ILP solver and an heuristic approach.

Obtained network is finally optimized in section 4.3, such that each user enjoys a *fair* amount of network resources, according to the *Nash arbitration scheme*. This optimal operation point is then compared with legacy strategies, namely *proportional fairness*, in order to assess whether this new proposed technique improvement is significant or not.

4.1 Geographical analysis

In order to obtain a decent map of the city, the two OpenStreetMap datasets corresponding to buildings and streets of the entire state of North Rein-Westphalia were downloaded, merged and cropped to remove anything outside Aachen border [25].

After those operations, the remaining 58,305 constructions and 9,759 roads were paired with the population density information and used to draw the map in Fig. 4.1.

As can be appreciated in the plot, the data is indeed quite accurate and suitable for the conversion to an abstract graph with streets as edges and road crossing

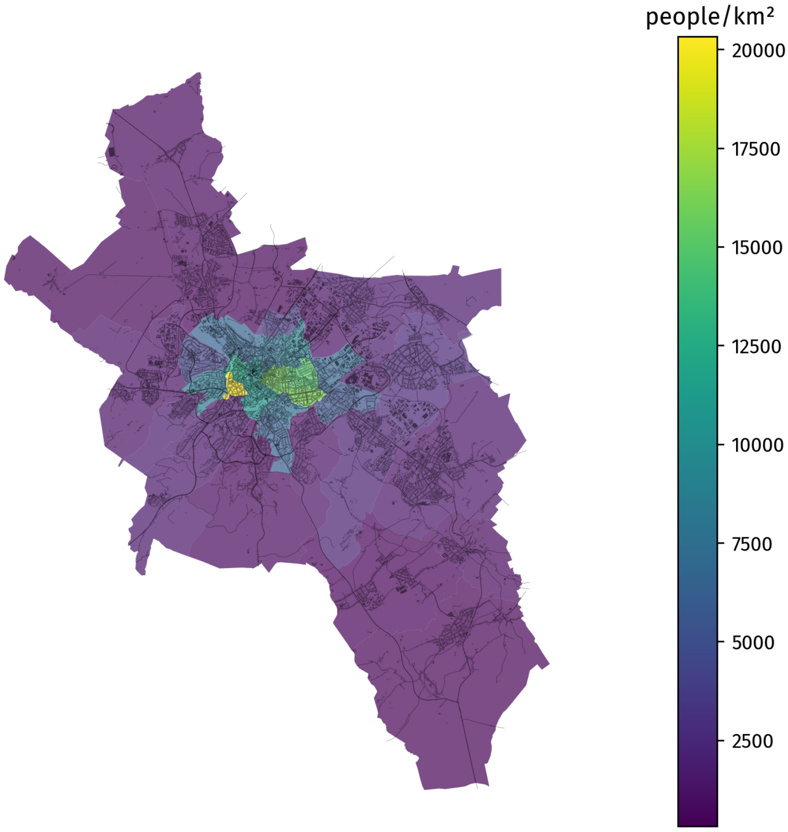


Figure 4.1: Summary of all Aachen district information we will employ: building, roads positions and population distribution.

as vertices. This procedure was performed using a dedicated library that took care of all the quirks of OpenStreetMap measures.* For example polygons were considered in contact up to a certain tolerance distance and everything outside the main connected component was pruned. Moreover all nodes close to each other less than 20m were merged: this removed many useless details and lowered the number of variables in the upcoming analysis.

Then each building was assigned residents, i.e. users of our access network, uniformly with respect to the area population density and building surface, as was explained in section 3.1.

To integrate customers information into the road graph, a first attempt was made where a new vertex was created for every building, but the number of variables turned out to be too high: it was then made the decision to assign the inhabitants of the city to the closer road crossing.
All long roads were split in segments, forced to be shorter than 200m: this way the average displacement introduced via this approximation was reduced to just 50m, tolerable for our purposes.

At the end of this pre-processing phase, the graph is made of 7,231 vertices and 9,272 edges and its complexity can be handled by our algorithms.
A visual representation is given in Fig. 4.2 that shows the result of a small part of the city center, as tiny details could not be otherwise discerned.

4.2 Network design

As was introduced in subsection 3.2.2, the design procedure is performed starting from the edge of the network, first positioning DSLAMs, then second level routers routers and finally the mainframe.

While the mathematical formulation is the same, each iteration requires different values for the problem parameters. Table 4.1 collects them all omitting the unnecessary ones, such as the fixed cost of the single mainframe which is not relevant in our analysis.

It is worth mentioning that the cost per unit c_r is split into two addends, accounting for the physical device and its connection to the mainframe. The price and the number of ports of the switching units match the most popular items in the market and industry best practices [26].

As anticipated in section 3.2 the exact solution to the placement optimization problems could not be obtained using ILP. Even with a commercial software such as CPLEX [31], in fact, computational time and memory demand exceeded

*See http://xiaming.me/posts/2016/12/18/process-gis-shapefile-with-graph-tools/

Figure 4.2: City topology is converted into an abstract graph.

Parameters	DSLAM	Routers	Mainframe
n_M [unit]	48	400	-
d_M [m]	1,500	-	-
c_r [€/unit]	1,000 + 30,000	15,000 + 85,000	-
c_f [€/m]	3	3	3
c_e [€/m]	100	100	100

Table 4.1: Values for problem parameters in the first two iterations.

all resources available.

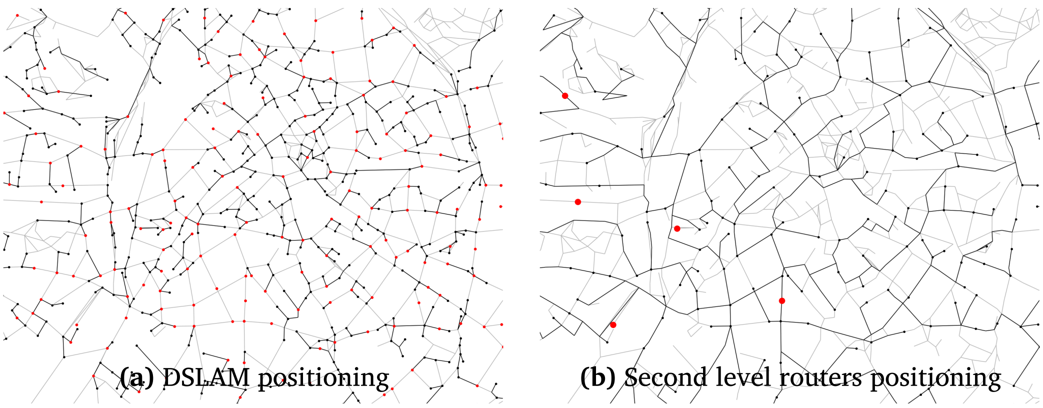
Although not conclusive, the solver provided useful insights on the valid solution domain, specifically a lower bound for the objective function, obtained with the countinous relaxation of problem (3.2).

These limits are then compared against the configuration obtained via heuristic algorithm, presented before in subsection 3.2.4.
Table 4.2 clearly shows that the heuristic result is indeed remarkably close to the theoretical optimum and proves that the choices and approximations made previously indeed captured all relevant features of the problem.

Problem	DSLAM	2nd level routers
Number of groups	1,125	72
ILP cost lower bound [M€]	65.05	38.08
Heuristic cost [M€]	67.73	39.38
Heuristic gap	4%	3%

Table 4.2: Cost of heuristic solution is compared to the theoretical limit given by ILP.

A visual representation of the obtained clusters, groups of devices connected to the same switching unit, is given in Fig. 4.3 and in Fig. 4.4. Again the map is cropped in order to scale at the proper level of detail.



(a) DSLAM positioning **(b)** Second level routers positioning

Figure 4.3: The root nodes in red are hubs for terminals, black points.

Figure 4.4: The mainframe, red dot, is located in an industrial complex and is connected to all second level routers.

4.3 Network optimization

4.3.1 Actual chapter

As mentioned in section 3.3, in our experiments we study the behaviour of our reference access network for different bandwidth demands and service profiles.

More specifically, former point is taken care of via p_{active} parameter, that is the probability a certain user is communicating or not. In order to describe the latter factor, instead, applications are randomly split into video streaming and web page browsing, according to a fraction $p_{streaming}$.

Two different allocation strategies are evaluated, traditional proportional fairness and the more sophisticated Nash arbitration scheme, computed via the *heuristic* Algorithm 3.3.

With respect to global objective function $f(\rho)$, defined in (3.14), our proposed solution indeed provides a more fair operation point than the legacy one, proving that even an approximation of the optimum outperforms what currently done in these kinds of scenarios.

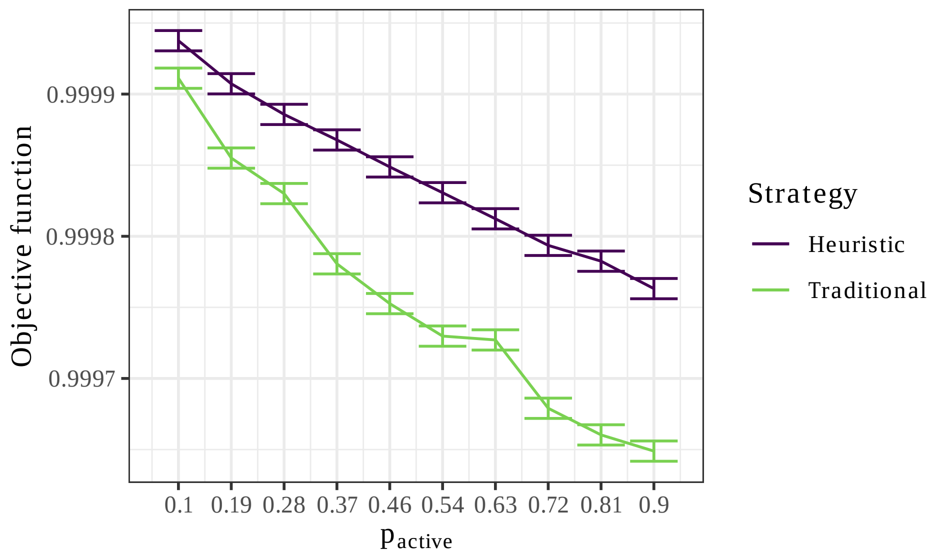


Figure 4.5: Performance for $p_{streaming} = 0.1$, with 95% confidence intervals.

This gap can be easily spotted in Fig. 4.5, Fig. 4.6 and Fig. 4.7, where average performance across multiple runs is computed for different values of p_{active} and $p_{streaming}$.

Moreover, this difference is more evident when offered traffic is higher, i.e. when the situation is more difficult to handle from the point of view of the

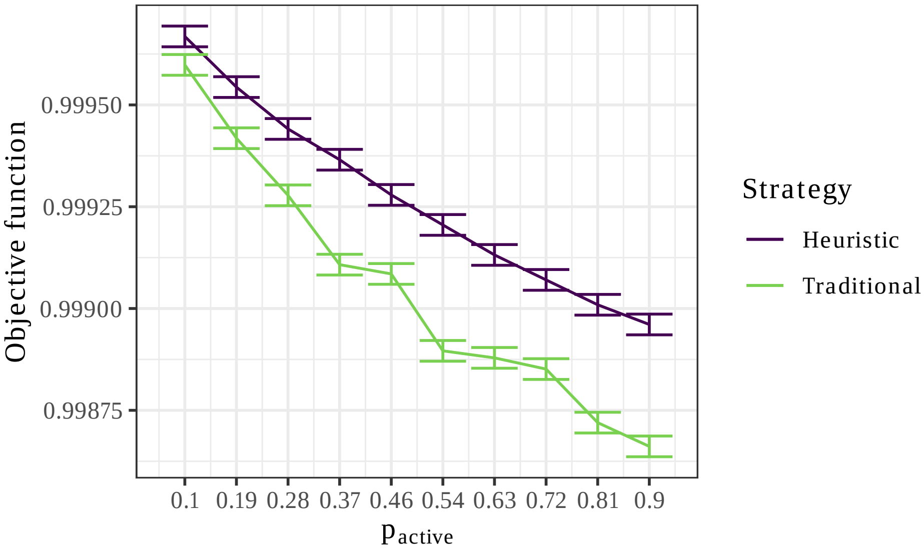


Figure 4.6: Performance for $p_{streaming} = 0.5$, with 95% confidence intervals.

administrator. This observation suggests that this novel approach can be relevant from a practical point of view as the infrastructure size and complexity, and thus its cost, is often dictated by worst-case scenarios.

As a final remark on these results, proposed method performance is smoother, somewhat more predictable: again is a strong point in the field, as it is more robust to system parameters estimation errors.

We then analyze utility distributions for the two techniques, focusing on extreme values of p_{active} and $p_{streaming}$.

When offered traffic is lower, as shown in Fig. 4.8, it is apparent how realizations of utility are overall closer to one, certifying the small, but noticeable, gain seen before for the novel approach.

When instead in Fig. 4.9 the number of active users increases the situation is trickier to interpret: again though the proposed solution tends to move towards the optimal point.

Despite all these promising results of proposed heuristic approach, it is currently only a lower bound on the theoretical optimum, exact solution of the mathematical problem. So, there is room for progress in the search of the best working point.

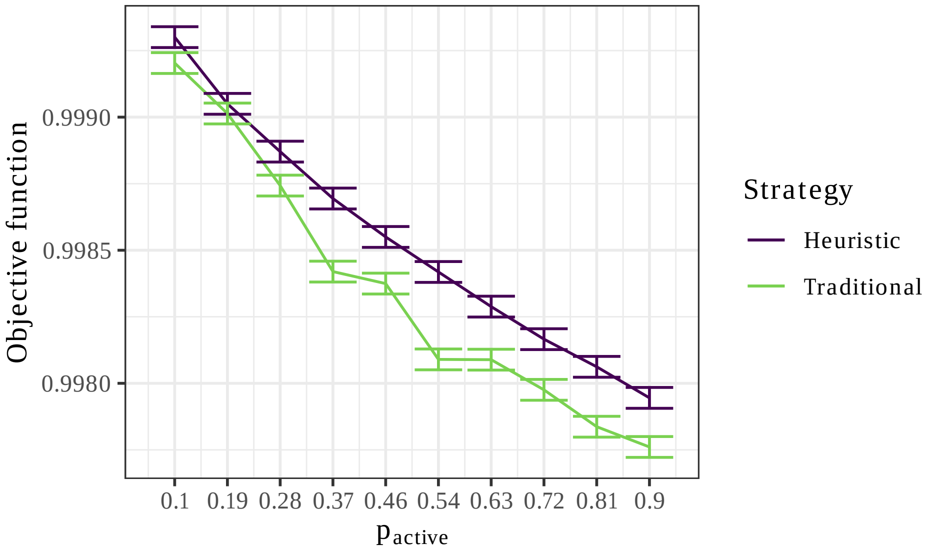


Figure 4.7: Performance for $p_{streaming} = 0.9$, with 95% confidence intervals.

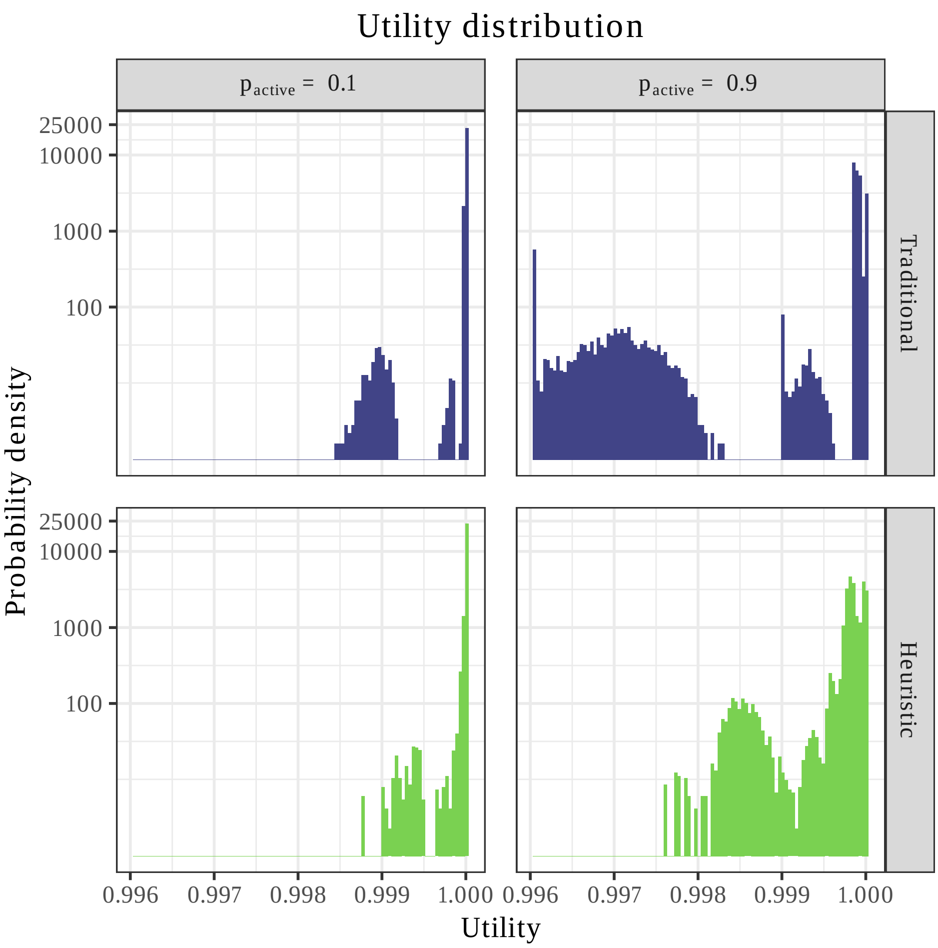


Figure 4.8: Utility distribution for $p_{streaming} = 0.1$.

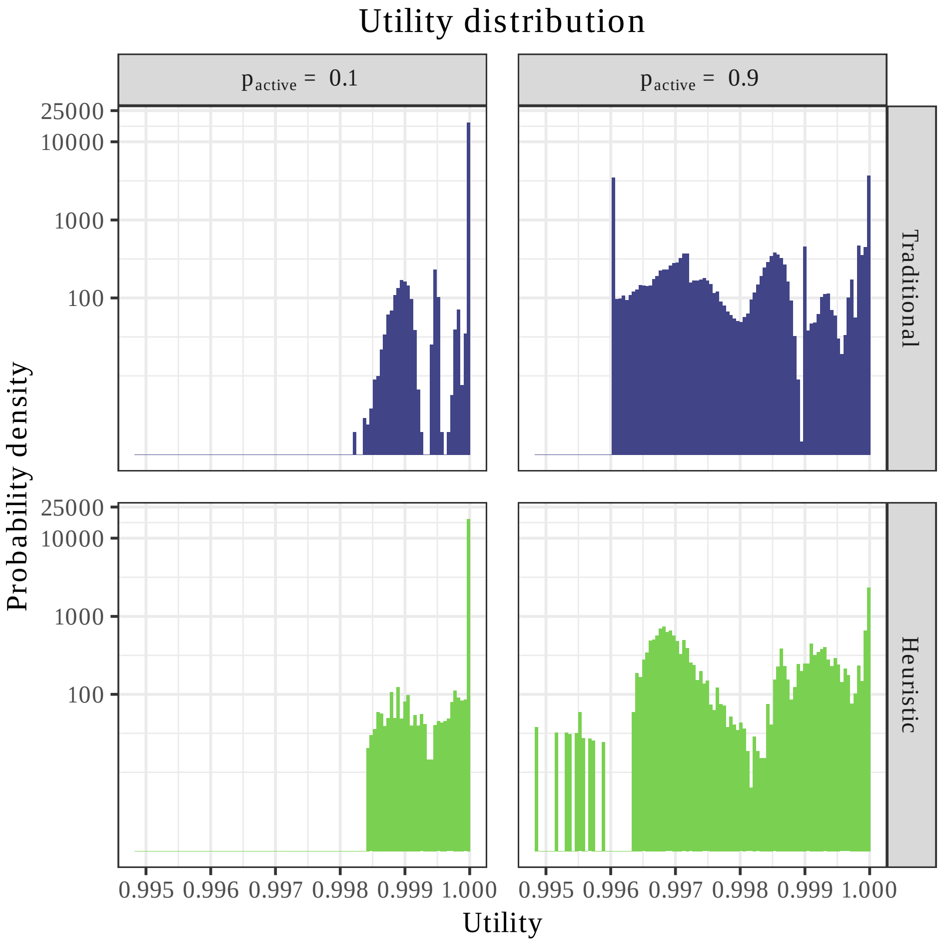


Figure 4.9: Utility distribution for $p_{streaming} = 0.9$.

5
Conclusions

This thesis discusses how to design, tune and optimize an access network from scratch, using only topographic and demographic information.

The end-to-end approach presented, meaning mathematical models and algorithms, can be applied to a generic problem as no particular assumption were taken about our case of study, the city of Aachen. Here we design a traditional access network for an Internet Service Provider (ISP), but the same techniques can be used, for example, to implement emergency communication infrastructure for areas subjected to natural disasters. In both situations, in fact, a number of people spread over a geographical area need to be connected to the main Internet. Either for business or pleasure or for calling rescue teams and families, the problem can be treated in a similar way.

Considering the first step, infrastructure design, we can assert that our model is more complete and our solutions better than what is available in literature at the moment. We can in fact consider properly all technological limitations of an FTTH network and allow for ad-hoc considerations in the optimization process. Moreover, our bounds with respect to the optimum are narrower than any state-of-the-art approach.

Once built the network, Software Defined Network (SDN) principles are applied to its management, entrusting packet flow control to a central unit.

Bandwidth distribution is directed by this *controller* according to a game theoretical framework, called *Nash arbitration scheme*. This ensure the final configuration to balance packet flows and meet all desirable assumptions of *fairness*.

Utility functions, needed to fully characterize the allocation game, are application specific and estimate actual Quality of Experience (QoE). This is a step forward with respect to common Quality of Service (QoS) metrics, as this way we maximize actual customer satisfaction and tailor service for end user peculiarities.

Indeed this strategy pays off, as improvement with respect to the legacy *proportional fairness* strategy is measurable and significant in terms of performance and reliability.

As proved in this thesis, the adopted approach is indeed promising: future development could consider using more connection metrics, such as delay and jitter, since bandwidth alone cannot completely describe QoE for some applications. These works could therefore assess whether a trade-off can be made among these quantities to better fit user requirements.

Moreover the Nash arbitration scheme, here only approximated, could be computed exactly or, if not possible, approximated via better and more specific heuristic algorithms.

References

[1] M. Andrews and L. Zhang, "The access network design problem," in *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, Palo Alto, CA, 1998, pp. 40–49.

[2] Alphabet Inc., "Google video quality report," https://www.google.com/get/videoqualityreport/, accessed: 2019-03-09.

[3] Cisco VNI Forecast, "Cisco visual networking index: Forecast and trends, 2017-2022," *Cisco Public Information*.

[4] A. Hakiri, A. Gokhale, P. Berthou, D. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Computer Networks*, vol. 75, 2014.

[5] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano *et al.*, "Jupiter rising: A decade of clos topologies and centralized control in Google's datacenter network," *ACM SIGCOMM computer communication review*, vol. 45, no. 4, pp. 183–197, 2015.

[6] S. Voß, "Steiner's problem in graphs: Heuristic methods," *Discrete Applied Mathematics*, vol. 40, no. 1, pp. 45–72, 1992. [Online]. Available: https://doi.org/10.1016/0166-218x(92)90021-2

[7] D. Rehfeldt, "A generic approach to solving the Steiner tree problem and variants," Master's thesis, 2015.

[8] T. Koch and A. Martin, "Solving Steiner tree problems in graphs to optimality," *Networks*, vol. 32, no. 3, p. 207–232, Oct 1998. [Online]. Available: http://dx.doi.org/10.1002/(sici)1097-0037(199810)32:3<207::aid-net5>3.0.co;2-o

[9] M. Leitner, I. Ljubic, M. Luipersbeck, M. Prossegger, and M. Resch, "New real-world instances for the Steiner tree problem in graphs," in *Technical report, Technical report, ISOR*, 2014.

[10] A. Mitcsenkov, G. Paksy, and T. Cinkler, "Geography- and infrastructure-aware topology design methodology for broadband access networks (FTTx)," *Photonic Network Communications*,

vol. 21, no. 3, pp. 253–266, Jun 2011. [Online]. Available: https://doi.org/10.1007/s11107-010-0297-4

[11] T. Benson, A. Akella, and D. Maltz, "Unraveling the complexity of network management," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'09, Boston, Massachusetts, 2009, pp. 335–348. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558977.1559000

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, p. 69, 2008. [Online]. Available: https://doi.org/10.1145/1355734.1355746

[13] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, Jul. 2008. [Online]. Available: http://doi.acm.org/10.1145/1384609.1384625

[14] S. Jain, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, A. Vahdat, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, and J. Zhou, "B4: experience with a globally-deployed software defined WAN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013. [Online]. Available: https://doi.org/10.1145/2534169.2486019

[15] WMware Inc., "VMware NSX virtualization platform," https://www.vmware.com/products/nsx.html, accessed: 2019-03-09.

[16] Linux Foundation, "Opendaylight: A linux foundation collaborative project," https://www.opendaylight.org/, accessed: 2019-03-09.

[17] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," IETF, Tech. Rep., 2009.

[18] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using OpenFlow-assisted adaptive video streaming," in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. Yasumoto International Academic Park, The Chinese University of Hong Kong: ACM, 2013, pp. 15–20.

[19] K. u. R. Laghari, O. Issa, F. Speranza, and T. H. Falk, "Quality-of-Experience perception for video streaming services: Preliminary subjective and objective results," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Hollywood, CA, 2012, pp. 1–9.

[20] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: Where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, vol. 52, no. 2, 2011.

[21] C. Douligeris and R. Mazumdar, "On Pareto optimal flow control in an integrated environment," in *Proc. of the 25th Allerton Conference on Communication, Control and Computing*. Allerton, Illinois: IEEE Communications Society, 1987.

[22] R. Mazumdar, L. Mason, and C. Douligeris, "Fairness in network optimal flow control: Optimality of product forms," *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 775–782, 1991. [Online]. Available: https://doi.org/10.1109/26.87140

[23] S. Tadelis, "Game theory, an introduction," *Economic Record*, 2013.

[24] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, *IEEE Journal on Selected Areas in Communications*.

[25] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," https://www.openstreetmap.org, accessed: 2019-04-03.

[26] *Layer 2 WAN - Technology Design Guide*, https://www.cisco.com/c/dam/en/us/td/docs/solutions/CVD/Aug2014/CVD-Layer2WANDesignGuide-AUG14.pdf, Cisco Systems Inc., August 2014.

[27] M. Diané and J. Plesník, "An integer programming formulation of the Steiner problem in graphs," *Mathematical Methods of Operations Research*, vol. 37, no. 1, pp. 107–111, 1993.

[28] R. Schatz, S. Egger, and A. Platzer, "Poor, good enough or even better? bridging the gap between acceptability and QOE of mobile broadband data services," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, June 2011, pp. 1–6.

[29] C. Xing, Y. Jing, S. Wang, S. Ma, and H. V. Poor, “New viewpoint and algorithms for water-filling solutions in wireless communications,” *arXiv preprint arXiv:1808.01707*, 2018.

[30] P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing,” in *Simulated annealing: Theory and applications*, 1987, pp. 7–15.

[31] IBM, “IBM ILOG CPLEX Optimization Studio,” https://www.cplex.com, 2017, accessed: 2019-04-03.