Budapest University of Technology and Economics
Doctoral School of Informatics
Department of Telecommunications and Media Informatics

OPTIMAL RESOURCE POOLING OVER LEGACY EQUAL-SPLIT LOAD BALANCING SCHEMES

Krisztián Németh

M.Sc. in Computer Science Engineering

Summary of the Ph.D. Dissertation

Supervised by

Gábor Rétvári, Ph.D.
Senior Research Fellow

Budapest, Hungary
2018

1 Introduction

Treating separate network resources as one and sharing it among users is a technique inherent to the Internet. This scheme, often called the Resource Pooling Principle [1], can be observed at several aspects of today's networks. Examples of this principle include multipath routing, multihoming, Ethernet Link Aggregation Groups, load balancing between application level servers (such as web-servers or database servers), load balancing in Traffic Engineering. Content Delivery Networks are also a form of resource pooling, just as cloud storage and cloud computing. To realize these services, data centers are being installed rapidly, often utilizing parallel paths, which are, in many of the cases, asymmetric in capacity [2]. Furthermore, several new concepts, such as network virtualization and Software Defined Networking (SDN) appeared in the recent years, which also take advantage of the pooling principle in order to optimally exploit the network resources.

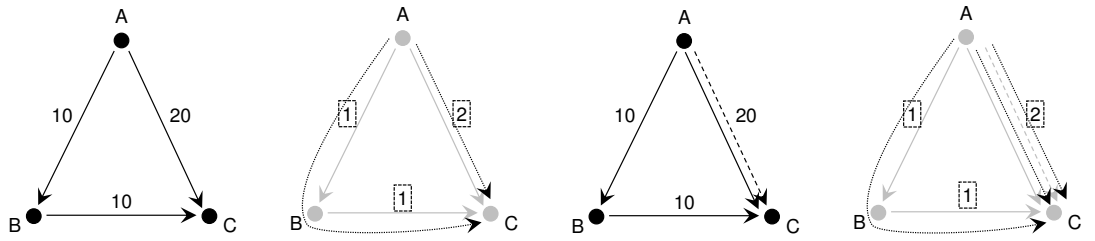
This list is far from being comprehensive, yet it shows the versatility of scenarios where resources are pooled. There are several reasons to do so. First, its inherent redundancy increases the robustness against component failures. Second, by dynamically allocating more resources for a temporal peak usage higher level services can be offered on the same infrastructure, utilizing statistical multiplexing. Third, having a greater freedom to couple demands and resources more efficient network utilization can be achieved along with a more scalable service.

The implementation of resource pooling, however, is challenging as the load balancers can often split the incoming demands only roughly equally amongst the resources. As an illustration, a load balancer between two web-servers typically splits the incoming requests in half, which heavily hinders the overall performance if one of the back-end servers are for instance twice as powerful as the other. Likewise, in routing protocols such as OSPF [3] or IS-IS [4] Equal-Cost Multipath (ECMP) is used to distribute the traffic over the shortest paths with the same cost[1]. ECMP, however, is only able to split traffic between these paths uniformly, even if they have different capacities, which poses a giant barrier when aspiring to an optimal Traffic Engineering [2, 5, 6, 7].

There are several existing proposals which target to improve specific cases of this issue. Weighted Cost Multipathing (WCMP, [2]), for example, aims unequal traffic splitting at data centers. It assumes SDN-capable switches, and operates by replicating rule table entries. Niagara [6] is another SDN-based proposal, which provides flexible traffic splitting between load balancers by building SDN rules based on the last bits of the source address. Fibbing [8] is another interesting architecture, which promises centralized control over distributed routing, without SDN. It works by effectively "lying" to OSPF, advertising fake nodes and links through standard routing-protocol messages. A recent application of Fibbing directly targets load-balancing [9]. These proposals, however, are more or less coupled to a single field of application and they either require a currently not widely deployed technology (i.e., SDN), or, in the case of Fibbing, would introduce a new level of abstraction, which it is yet unclear if operators are willing to cope with.

As a solution, I introduce a technique called *Virtual Resource Allocation* (VRA) to realize optimal resource pooling over legacy equal-split load balancing schemes. The basic

[1] or same length, which is identical in this case



(a) Original topology with link capacities (b) Link weights and shortest paths (c) Adding a virtual link (d) Link weights and shortest paths

Figure 1: A triangular network. Demand: $A \to C : 30$

idea of VRA is to virtually multiply the available parallel resources so that the load balancing system sees a greater number than what actually exists. The virtual resources are then grouped and assigned to the physical ones, thereby tricking the legacy equal splitting technology into approximating the required non-equal load division over the existing media.

To continue the previous example, one can install two virtual machines on the more powerful web-server and present them, along with the unmodified less powerful server, to the load balancer. It then sees three servers, and by realizing equal split between them the higher capacity one will eventually end up with 2/3 part of the total load, as desired.

The following example for the Virtual Resource Allocation concept is from the topic of Traffic Engineering. Consider the triangular network shown in Fig. 1(a). Suppose we would like to transfer 30 units of traffic from A to C without overutilizing any of the links. Using stock OSPF would allow us to set the link weights[2], thereby we could easily create two equal cost shortest paths (i.e. paths with minimal total cost/weight): $A - B - C$ and $A - C$, by using for example the weights shown in Fig. 1(b). On the other hand, OSPF ECMP only allows splitting the traffic equally between the shortest paths implying a 150% load on links $A - B$ and $B - C$.

If, however, we could set up a virtual link on top of the existing link $A - C$ and expose it to OSPF (see Fig. 1(c)), it would happily split the traffic in three, sending one third on path $A - B - C$ and the rest on physical link $A - C$ (Fig. 1(d)). Naturally, installing a virtual link over physical link $A - C$ does not change its capacity, it only enables OSPF ECMP to use its full potential. The link weights would also remain unchanged, and the new virtual link would have the same weight as the respective physical one. By this simple administrative intervention we can route the traffic through this network without exceeding the link capacities.

One advantage of my VRA proposition is that it is incrementally deployable, since it is perfectly fine to set up virtual resources only at a subset of the network nodes. Moreover, unlike most other proposals, VRA is fully compatible with existing hardware or software components in the network. Finally, VRA is extremely efficient, as my numerical results indicate that by adding only a small set of virtual resources the ideal traffic split ratio can be very well approximated, resulting in substantial performance gain.

[2]Link weights are also often called link costs, link metrics or SPF (Shortest Path First) metrics. I will use these terms interchangeably.

2 Research Goals

Throughout this work Traffic Engineering (TE) in IP networks [10] is used to describe the VRA proposal. The idea is to set up virtual links alongside the existing ones and present them to OSPF so that near-optimal TE can be achieved without any hardware or software modification on the network infrastructure. Let me emphasize, however, that TE is just a descriptive example application of the VRA concept, and its possible fields of usage are much broader. To name one other use case, in certain SDN-based scenarios VRA can be used for rule table optimization.

There are several general approaches to provide near-optimal TE and in this work I address three of them. In the first one, which I call *Overlay Optimization*, it is possible to set up end-to-end tunnels with MPLS-TE [10] for example. I also assume that several tunnels can coexist between a source and a destination endpoint pair for the sake of better resource utilization. Using the VRA concept these tunnels can be virtually multiplied and then OSPF ECMP is deployed on top of this overlay. The goal here is to find the best approximation of a predefined traffic split ratio between the paths connecting a given source and destination pair using only a limited number of virtual paths.

In the other two proposed TE approaches, which I call *Peer-Local Optimization* and *Peer-Global Optimization*, there is no overlay network and traffic optimization takes place directly on the physical infrastructure. Furthermore, here I do not assume the presence of any advanced TE methods, only OSPF is used for TE purposes. This is called OSPF Traffic Engineering (OSPF-TE), where the basic idea is to adjust the administrative link costs so that the shortest paths calculated by OSPF will map exactly to the ones chosen by the administrator [11, 12].

I have extended the classical OSPF-TE concept by setting up virtual links parallel to existing physical ones.[3] The engineering problem to solve in this case is to determine the link weights and the number of parallel virtual links for each physical link in a network domain so that a predefined metric is optimized. As its name implies, in Peer-Local Optimization these decisions are made locally at the network nodes, while in Peer-Global Optimization it happens in a centralized fashion at the network level. In this work the considered measure is the widely accepted maximal link utilization (MLU), which is minimized over all the links of the domain. Link utilization is defined the usual way: link usage divided by link capacity.

The problem is challenging as it is known that finding the best link weight configuration for OSPF-TE (without virtual links) is NP-hard by itself [5] and even approximating it by a computationally efficient algorithm within any constant ratio is infeasible [13].

3 Research Methodology

I have chosen the analytical approach to address my research goals as the results gained this way are generally more universal and well-founded than the ones obtained by measurement or simulation. In general the downside of the analytic method is that a large set of problems

[3]There are several ways to set up virtual links, but the exact method is out of the scope of this work, as I only focus on the effect of the virtual links on the network performance.

are too complex to be handled this way, but in this case it was possible to deal with this complexity. Consequently, a large part of my thesis claims are based on theorems and mathematical proofs.

In a few of my other theses I propose algorithms to solve some given problems. In these cases I have determined the computational complexity of the algorithms by analytical examination. Many of my algorithms are based on solving Linear Programs or (Mixed) Integer Linear Programs. Some of my theses contain NP-completeness proofs, for which I have used the customary Karp-reduction.

To see how well the proposed algorithms perform in realistic environments, I have implemented a simulation framework, which was used to compare the algorithms against each other and the current best-practice method (TOTEM Interior Gateway Protocol Weight Optimization Tool, [14, 15]). I have implemented the framework and the optimization algorithms in C++ using the LEMON Graph Library [16]. I have solved the embedded linear programs using the IBM ILOG CPLEX Optimizer [17]. The results gained this way give a valuable insight into the performance potential of my proposed optimization techniques.

4 New Results

4.1 Virtual Resource Allocation and Overlay Optimization

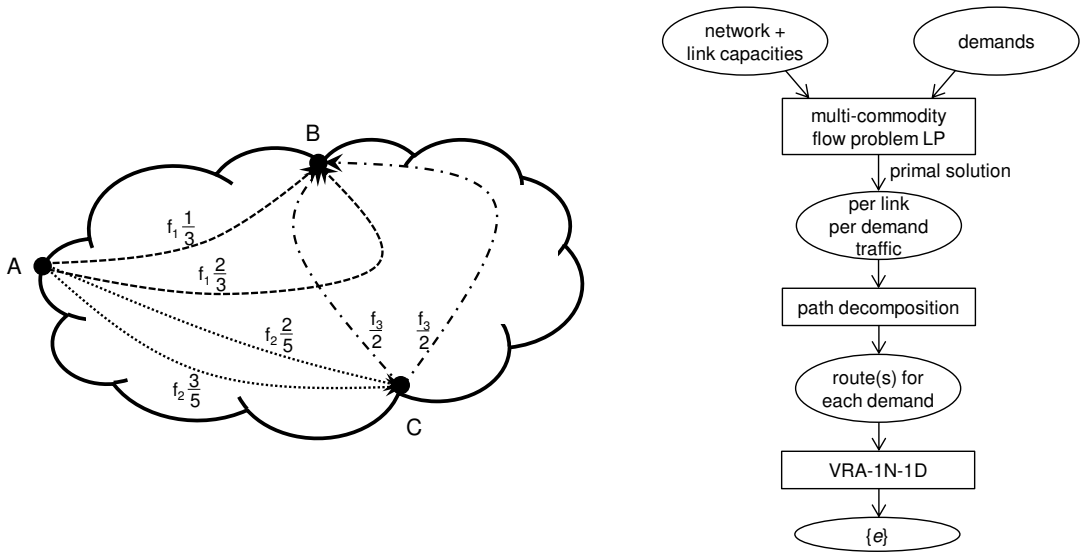
Thesis Group 1. [C3, J1] *I have studied the possibility of enhancing load balancing schemes by unequal traffic splitting when the underlying technology only offers uniform data distribution among the resources. For this I have proposed the Virtual Resource Allocation technique that augments the load balancer to realize an almost arbitrary traffic split ratio. For the OSPF Traffic Engineering application scenario I have introduced and analyzed the Overlay Optimization method, which is a specialization of the Virtual Resource Allocation, utilizing an overlay network.*

The concepts of Virtual Resource Allocation and OSPF Traffic Engineering has been introduced in the first two chapters of this work. This first thesis group focuses on the *Overlay Optimization* method, as defined below.

4.1.1 Definition and Attributes

Thesis 1.1. *I have proposed a solution, named Overlay Optimization, to the Traffic Engineering problem in communication networks that uses end-to-end tunnels with parallel virtual paths and OSPF routing on top of this overlay network. As a part of this solution, I have formalized the Virtual Resource Allocation problem for only one network node and one demand as an optimization problem. I have shown via an example that the performance of OSPF Traffic Engineering can be enhanced by Overlay Optimization.*

A sample scenario for Overlay Optimization is plot in Fig. 2(a). In this simple transit network there are three edge routers A, B and C, and a full mesh MPLS overlay is realized between them containing two paths per router pair. This MPLS overlay, in turn, is seen as



(a) Overlay Optimization example

(b) Overlay Optimization with tunnel/route setup

Figure 2: Overlay Optimization

an IP topology deployed on top, which runs plain OSPF as a routing protocol. Easily, if the ideal traffic splitting ratios are like the ones given in the figure then this traffic allocation is impossible to implement with ECMP. With my proposed technique, however, we can set up 4 virtual links (one between $A-B$ and three between $A-C$) to obtain exactly the required splitting.[4]

Overlay Optimization can also be used in the more general case, when only a capacitated network and the demands are given, and we can assume the ability of setting up (possible parallel) end-to-end tunnels. In this case we first have to calculate a set of end-to-end tunnels, then use the VRA Overlay Optimization method over these paths. The major steps of my proposed technique are shown in Fig. 2(b). For now the most important step is VRA-1N-1D (meaning VRA for One Node, One Demand). In this case splitting only occurs at the source nodes, which greatly simplifies the analytical treatment of the problem, as a VRA problem can be decomposed into D independent VRA-1N-1D problems, where D is the number of demands.

Let me now define the VRA-1N-1D problem formally. We are given a single node, where the traffic of a single demand has to be split. It is supposed to use k outgoing links (or paths/tunnels, but for simplicity I will use "link" in the remainder of this section), each with g_1, g_2, ..., g_k desired traffic volume (see Table 1 for a list of notations). Furthermore let $G_0 = \sum_{i=1}^{k} g_i$. Our objective is to share the traffic over the outgoing links using OSPF ECMP such that the actually emerging h_1, h_2, ..., h_k subflow values are as close as possible to the nominal g_1, g_2, ..., g_k subflow volumes. Here "closeness" between the ith subflows is

[4]The phrase "virtual path" could be more appropriate in this case, but for simplicity I continue to call them virtual links.

Notation	Description
k	number of outgoing links used
$g_1, g_2, \ldots, g_k \in \mathbb{Z}^+$	desired traffic volume per outgoing links
$G_0 = \sum_{i=1}^{k} g_i$	total traffic volume
$h_1, h_2, \ldots, h_k$	actual traffic volume per outgoing link
$U_i = h_i/g_i$	error on the ith outgoing link
$U = \max_i U_i$	error of a virtual resource allocation
$e_1, e_2, \ldots, e_k$	number of allocated links (physical and virtual together)
$E = \sum_{i=1}^{k} e_i$	total number of allocated links
$Q \in \mathbb{Z}^+$	upper bound on the total number of links

Table 1: VRA-1N-1D notation summary

defined as the per link error $U_i = h_i/g_i$, and the ultimate error metric to be minimized (U) is the maximum of the per link errors.[5]

To reach our objective, I apply virtual links parallel to the physical ones. Let e_i denote the total number of (virtual and physical) links in place of the ith physical link, and $E = \sum_i e_i$ the total number of allocated links. To save space, I only examine the case without link disabling possibilities (i.e. $e_i > 0$). Applying the equal-split principle of OSPF ECMP we get: $h_i = G_0 e_i / \sum_{j=1}^{k} e_j = G_0 e_i / E$, $i = 1 \ldots k$ and $U = \max_i h_i/g_i = \max_i G_0 e_i/(E g_i)$. Furthermore, I suppose that the total number of outgoing links for a demand is limited (just like in the practical router implementations), which I model by requiring $E \leq Q$.

The formal definition is:

Problem 1, VRA-1N-1D. Given k, $\{g_i\}$ and Q, find $\{e_i\}$ that minimizes U such that $\sum_i e_i \leq Q$.

To show the utility of Overlay Optimization, let us consider Fig. 1 again. Having path $A-B-C$ and two parallel paths $A-C$ will result in maximal link utilization of 1.0, while the best that can be achieved with plain OSPF-TE is a MLU of 1.5.

Thesis 1.2. *I have given bounds on the error of the VRA-1N-1D problem under different constraints.*

I have proven the following theorems about the error of VRA-1N-1D:

1. $U \geq 1$.
2. If $Q \geq G_0$ then $\exists\{e_i\}$ for which $U = 1$.
3. $U \leq G_0 / \min_i g_i$.
4. If G_0 is unbounded and E is bounded by a finite Q then U can be arbitrarily high for any $Q > 2$.

[5]Note that I refer to U_i and U as "errors", but in fact they represent actual-to-required traffic ratios. Usually zero or close-to-zero errors are preferred, but in this case $U = 1$ is the ideal condition.

The first statement is a lower bound. The second shows that if the number of usable virtual links is large enough then this lower bound can indeed be reached. The third statement is an upper bound as a function of the total traffic volume and the fourth is a negative result regarding the existence of a universal upper bound.

Although the existence of provable error bounds are important by itself, they are also used in the binary search algorithm at the next thesis.

4.1.2 Optimal Solution

Thesis 1.3. *I have given an optimal solution with pseudo-polynomial running time to the VRA-1N-1D problem. Furthermore, I have given an optimal, pseudo-polynomial time algorithm for the problem variant of minimizing the link number under a constraint on the maximal error. I have also given optimal, pseudo-polynomial time algorithms for variants of the previous two problems in which the error or link number minimization have to be done simultaneously at several nodes, while having a common constraint on the total link number or on the error, respectively.*

I have proven that the number of valid link allocations is $\binom{Q}{k}$ [Dissertation, Lemma 7]. This means for small Q values (like $Q \leq 30 \dots 50$) an exhaustive search may be feasible, but not for much larger ones. As VRA-1N-1D can be used in scenarios, where Q could be in the order of thousands (like SDN, where Q represents the maximal rule number), a more computationally efficient solution is necessary.

Algorithm 1.1 [Dissertation, Alg. 3.1] checks for a given α, k, $\{g_i\}$ and E whether or not it is possible to assign the links with $U \leq \alpha$. If the assignment is feasible then it also provides a solution and indicates if it is the only solution. I have proven that Algorithm 1.1 provides correct result and that it has a complexity of $O(k)$.

Next, I have proposed a binary search framework to find the minimal α for which there is a feasible solution of Alg. 1.1, given g_is and E [Dissertation, Alg. 3.2]. I have proven the correctness of the stop condition of its iteration loop and also that it runs in $\log(G_0^2 E)$ steps, yielding an overall $O(k \log(G_0^2 E))$ polynomial complexity.

What remains is to find the value of E that yields the smallest error subject to the given Q. This is done by Algorithm 1.2 [Dissertation, Alg. 3.3]. This is not a polynomial time algorithm as its complexity is $O(Qk \log(G_0^2 Q))$, which is not polynomial in the size of Q (i.e., $\log(Q)$). Yet, this complexity is low enough, so the algorithm is easily tractable for the practical use cases.

The next problem is to minimize the link number with a given maximal error.

Problem 2, VRA-1N-1D-Link-Min. Given k, $\{g_i\}$, and $U_{\text{lim}} \geq 1$, find $\{e_i\}$ that minimizes the total number of links (E) such that $U \leq U_{\text{lim}}$.

Consider Alg. 1.2, with $Q \leftarrow G_0$. The generated $U(E)$ is a weakly decreasing function, whose domain is a subset of the positive integers. The solution of the problem is E, where $U(E-1) > U_{\text{lim}}$ and $U(E) \leq U_{\text{lim}}$, or k, if $U(k) \leq U_{\text{lim}}$. The complexity is $O(G_0 k \log(G_0^3))$.

The remaining two problem variants tackle several optimizations simultaneously.

Problem 3, Parallel-VRA-1N-1D. Given k_n $(n = 1 \dots N)$, $\{g_{ni}\}$, and Q, find $\{e_{ni}\}$ that minimizes $U_{\max} = \max U_n$ such that $\sum_n E_n = \sum_n \sum_i e_{ni} \leq Q$.

Algorithm 1.1 VRA-1N-1D-Fixed-E

Input: α, k, $\{g_i\}$, E
Output: $feasible$, $single\_solution$, $\{e_i\}$

```
for i ← 1...k do
    x_i ← ⌊ α g_i E / G_0 ⌋
end for
if Σ_{i=1}^{k} x_i < E then
    feasible ← false
else if Σ_{i=1}^{k} x_i = E then
    feasible ← true
    single_solution ← true
else
    feasible ← true
    single_solution ← false
end if
if feasible = true then
    Solve the following set of equations to find an {e_i}:
        Σ_{i=1}^{k} e_i = E;  1 ≤ e_i ≤ x_i  (e_i ∈ ℤ, i = 1...k)
end if
```

Algorithm 1.2 VRA-1N-1D

Input: k, $\{g_i\}$, Q
Output: $\{e_i\}$, $U(E)$, U

```
best_U ← G_0 + 1.0
for E ← k...Q do
    {current_e_i}, current_U ← VRA-1N-1D-Bin-Search({g_i}, E)
    if current_U < best_U then
        best_U ← current_U
        {best_e_i} ← {e_i}
    end if
    U(E) ← best_U {used only in derivative algorithms}
end for
U ← best_U
{e_i} ← {best_e_i}
```

I have given a greedy algorithm that finds a solution to this problem [Dissertation, Alg. 3.4]. It uses the $U(E)$ function of each node, which is given by Alg. 1.2. I have proven that this algorithm provides an optimal result. I have also determined its complexity, which is $O(NQk \log(G_0^2 Q) + Q)$, where $k = \max k_i$, $G_0 = \max G_{0i}$.

Problem 4, Parallel-VRA-1N-1D-Link-Min. Given k_n $(n = 1 \dots N)$, $\{g_{ni}\}$, and U_{lim}, find $\{e_{ni}\}$ that minimizes $\sum_n E_n = \sum_n \sum_i e_{ni}$ such that $U_{\max} = \max U_n \leq U_{\text{lim}}$.

This problem can trivially be decomposed into N independent VRA-1N-1D-Link-Min problems, which can be solved as described above at Problem 2.

4.2 Peer-Local Optimization

Thesis Group 2. [C1, J1] *I have proposed and examined in detail another Virtual Resource Allocation scheme in the OSPF Traffic Engineering scenario that eliminates the overlay network from the architecture, thereby facilitating the deployment. This approach operates on the original network topology and makes decisions locally at the network nodes, so I named it Peer-Local Optimization.*

In the *Peer-Local Optimization* scenario we are given a capacitated network and a set of demands (see Fig. 3). The optimization task is to determine for each link a weight and the number of parallel virtual links, which, if fed together to OSPF, will result in minimal MLU (maximal link utilization). In other words, Peer-Local Optimization provides input for OSPF-TE [12] enhanced by VRA.

4.2.1 Definition and Attributes

Just as before, here we have a limit on the number of usable links per node as well, however, in this case the limit exists *per node per demand*, in line with the router constraint that a single traffic flow cannot be split onto too many outgoing links.

As an example consider the capacitated network in Fig. 4(a) with two demands: $A \to E : 30$, $A \to F : 40$. Clearly, for optimal routing all the links have to be fully utilized, requiring a traffic split of $2 : 1$ and $1 : 3$ for the demands at node A. Suppose we are allowed to use at most $Q = 4$ outgoing links per node per demand. We can reach an optimal solution by setting up virtual links as shown in Fig. 4(b). Although this way six links are leaving node A, none of the demands are split onto more than four, obeying the limit.

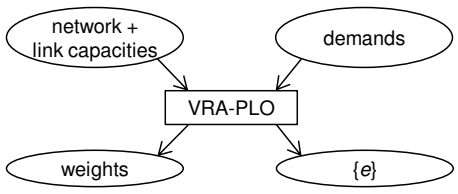
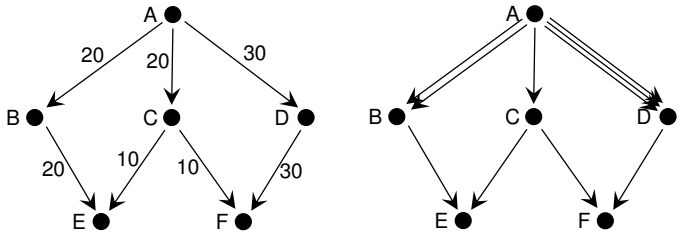


Figure 3: Virtual Resource Allocation–Peer-Local Optimization



(a) A network with link capacities

(b) Installed virtual links

Figure 4: Multi-demand constraint example. Demands: $A \to E : 30$, $A \to F : 40$

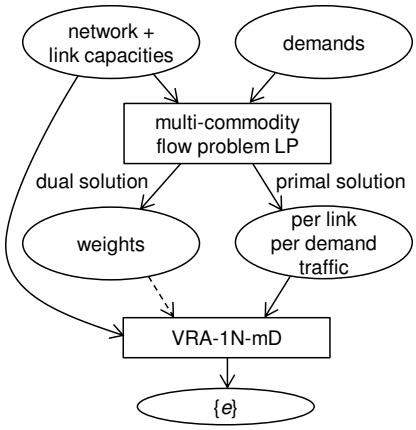


Figure 5: Peer-Local Optimization operation

Thesis 2.1. *I have proposed a new solution to the OSPF Traffic Engineering problem, named Peer-Local Optimization, which is not using overlays and relies solely on decisions made locally at the network nodes. As part of this solution I have formalized the Virtual Resource Allocation problem for one node and several demands as an optimization problem.*

The overview of the operation of Peer-Local Optimization is shown in Figure 5. The first step is the same as for Overlay Optimization: solving a multi-commodity linear program with splittable flows, which can be done in polynomial time. The primal solution provides the per link per demand traffic volumes and the dual corresponds to the link weights necessary for OSPF-TE. The final step is to solve the VRA-1N-mD problem for each node independently. VRA-1N-mD stands for "Virtual Resource Allocation for One Node and multiple Demands" and provides locally optimal virtual link settings.

The formal definition of the VRA-1N-mD problem, using the notations summarized in Table 2, is as follows.

Notation	Description
k	number of outgoing links used
D	number of demands
$G = (g_{ij}) \in \mathbb{Z}^{D \times k}$	desired traffic volume per demand per outgoing link ($g_{ij} \geq 0$)
$\Gamma = (\gamma_{ij}) \in \mathbb{R}^{D \times k}$	row-normalized version of matrix G
$\Sigma = (\sigma_{ij}) \in \{0,1\}^{D \times k}$	$\sigma_{ij} = 0$ if $g_{ij} = 0$, $\sigma_{ij} = 1$ otherwise
$G_i = \sum_{j=1}^{k} g_{ij}$	total traffic volume of demand i
h_{ij}	actual traffic volume per demand per outgoing link
e_1, e_2, …, e_k	number of allocated links (physical and virtual together)
$E_i = \sum_{j=1}^{k} e_j \sigma_{ij}$	total number of parallel links on shortest paths for demand i
$U_{ij} = e_j/(\gamma_{ij} E_i)$	per demand per link error (only where $\gamma_{ij} > 0$)
$U = \max_{\gamma_{ij}>0} U_{ij}$	per node error
$Q \geq E_i \ (\forall i), \ Q \in \mathbb{Z}^+$	upper bound on the number of usable links per demand

Table 2: VRA-1N-mD notation summary

For a network node A we are given a matrix

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{D1} & g_{D2} & \cdots & g_{Dk} \end{bmatrix} , \tag{1}$$

representing how demands $1 \ldots D$ arriving at the node should be split among outgoing links $1 \ldots k$: g_{ij} is the traffic volume that belongs to demand i and should be sent out on link j ($g_{ij} \in \mathbb{Z}$, $g_{ij} \geq 0$). We can assume for simplicity that G contains no all-zero rows or columns. Later on I will also use the row-normalized and the signum versions of G:

$$\gamma_{ij} = \frac{g_{ij}}{\sum_{n=1}^{k} g_{in}}; \ \ \sigma_{ij} = \begin{cases} 0, & \text{if } g_{ij} = 0 \\ 1, & \text{if } g_{ij} > 0 \end{cases} .$$

We set up e_j number of parallel links (including the physical and virtual ones) for outgoing link j of node A. I suppose that an existing link cannot be disabled (i.e., $e_j > 0$). Let $E_i = \sum_{j=1}^{k} e_j \sigma_{ij}$ be the total number of parallel links on the shortest paths for the ith demand and $G_i = \sum_{j=1}^{k} g_{ij}$ be the offered load for the demand. According to ECMP's equal-split rule, the per demand traffic volume on an outgoing link is: $h_{ij} = e_j G_i / E_i$. The per demand per link error is defined as the ratio of the transmitted traffic and the offered volume on a given outgoing link j, for a given demand i, but it is only defined if the offered traffic is non-zero: $U_{ij} = h_{ij}/g_{ij} = e_j/(\gamma_{ij} E_i)$, $\forall g_{ij} > 0$. The per node error (or shorty just error) is defined as the maximum of the per link per demand errors: $U = \max_{i,j:\gamma_{ij}>0} U_{ij}$.

To sum up, we can formulate the problem as follows:

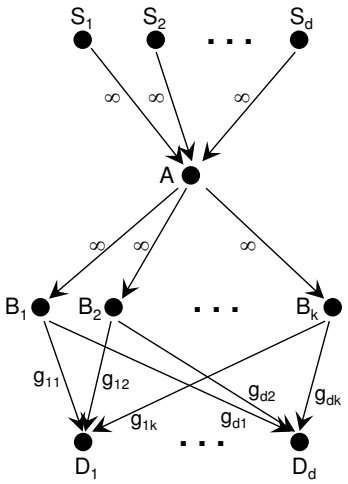


Figure 6: Capacitated network corresponding to a given matrix G

Problem 5, VRA-1N-mD. Given k, D, G, and Q, find $\{e_j\}$ such that $E_i \leq Q$ for all $i = 1 \dots D$ and U is minimal.

As a first approach, however, I will examine a simplified problem variant:

Problem 6, VRA-1N-mD-Unlimited. Given k, D, and G find $\{e_j\}$ such that U is minimal.

Note that in this latter case the problem is practically defined by matrix G only.

Thesis 2.2. *I have shown that matrix G, which forms the core of the VRA-1N-mD and the VRA-1N-mD-Unlimited problems, can be almost arbitrary: any nonnegative matrix with at least one non-zero element in each row and in each column can be matrix G for a given node in a suitable network.*

This thesis corresponds to [Dissertation, Theorem 11]. In the proof I construct a network for a given matrix G where after the Peer-Local Optimization process (see Fig. 5) at a certain node the required splitting ratios are exactly as given in G. Let G be in the form of (1). The corresponding capacitated network is shown in Fig. 6. There are altogether d demands in the system, the ith going from S_i to D_i and for each demand i the desired splitting ratio is given by g_{ij} $(j = 1 \dots k)$.

This result is significant as it allows us to focus on the matrices only, instead of the possibly much more complex networks.

Thesis 2.3. *I have given bounds on the error of the VRA-1N-mD and the VRA-1N-mD-Unlimited problems along with a polynomial time algorithm that decides whether the general lower bound can be reached for a particular problem instance with unlimited number of links.*

Regarding the error bounds of VRA-1N-mD and VRA-1N-mD-Unlimited I have proven the following statements [Dissertation, Lemmas 12–14]:

1. $U \geq 1$.
2. $U \leq (\min_{i,j:\gamma_{ij}>0} \gamma_{ij})^{-1}$.

3. There is no universal (G-independent) upper bound on the per node error.

These claims are valid for both the limited and the unlimited number of links version of the problem.

I call a matrix G *consistent* if and only if $U = 1$ can be achieved with a suitable $\{e_j\}$, allowing unlimited number of parallel links. I have given an algorithm that decides whether a given G is consistent, and if it is, it also supplies an $\{e_j\}$ for which $U = 1$ [Dissertation, Alg. 4.1]. Its complexity is $O(d^2k)$, meaning that it runs in polynomial time.

4.2.2 Unlimited Number of Links

Let us now examine Problem VRA-1N-mD-Unlimited, i.e., solving VRA-1N-mD with unlimited number of links. For simplicity, in this subsection I will use the normalized version of the link number e_j, denoted by f_j to avoid confusion: $f_j = e_j / \sum_{i=1}^{k} e_i$ ($f_j \in \mathbb{R}^+$, $\sum_{j=1}^{k} f_j = 1$).

Thesis 2.4. *I have proven that an optimal virtual link settings for VRA-1N-mD-Unlimited cannot always be reached using finite number of links.*

I have actually shown that there is at least one VRA-1N-mD-Unlimited problem, where matrix G contains integers only but the single optimal solution contains only irrational numbers as f_js [Dissertation, Theorem 16]. In the proof of this theorem I show that the problem given by matrix

$$G = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

has a single optimal solution:

$$f_1 = \frac{2}{5}(7 - \sqrt{34}),\ f_2 = \frac{1}{5}(-16 + 3\sqrt{34}),\ f_3 = \frac{1}{5}(7 - \sqrt{34})\ ,$$

which also proves the claim of this thesis.

Thesis 2.5. *I have shown that no algorithm can give an optimal solution to the VRA-1N-mD-Unlimited problem in finite number of steps; even if the number of steps may depend on the actual problem.*

A solution is given as $f_1, f_2, \ldots, f_k$, where $f_j \in \mathbb{R}^+$. We expect real constants f_j to be presented by an algorithm in some sort of closed form, but "closed form" can be defined in several ways. For now I require f_js to be given by finite expressions that consist of integer constants and the usual $+$, $-$, $\cdot$, $/$ and the nth root ($n \in \mathbb{Z}^+$) operators only.

This thesis is based on [Dissertation, Theorem 18], which states that there is at least one VRA-1N-mD-Unlimited problem, whose only optimal solution contains at least one f_j that cannot be written in a closed form as defined above.

The idea is that the optimal solution cannot be computed in finite number of steps if it cannot be written in a closed form, since writing the output is part of the solution. In the

proof I have examined the following matrix:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 & 0 & 0 \\ 6 & 6 & 1 & 0 & 0 & 0 \\ 6 & 6 & 6 & 1 & 0 & 0 \\ 6 & 6 & 6 & 6 & 1 & 0 \\ 6 & 6 & 6 & 6 & 6 & 1 \end{bmatrix} .$$

I have proven that for this matrix f_1 is the solution of the following polynomial equation:

$$923\,521 f_1^5 - 16\,980\,870 f_1^4 + 118\,664\,280 f_1^3 - \\ - 390\,577\,680 f_1^2 + 934\,673\,904 f_1 - 336\,117\,600 = 0 \ . \quad (2)$$

I have used the mathematical software Maple [18] to show that this polynomial has got a single real root only (and four complex ones). According to Galois theory [19], a polynomial equation can be solved by radicals[6] if and only if its Galois group is a solvable group. Using Maple I have found that the Galois group of the polynomial given in (2) is the symmetric group S_5. This group, consisting of 120 elements, is not solvable, meaning that (2) cannot be solved by radicals, which concludes the proof.

Thesis 2.6. *I have given an approximation algorithm that can find, in polynomial time, a solution that is arbitrarily close to the optimal solution of the VRA-1N-mD-Unlimited problem.*

As a first step, I set up Linear Program (LP) 2.1 [Dissertation, LP 4.2] that computes $\{f_j\}$ while keeping the per node error under a given constant α.[7] Naturally, for too small αs the LP will not have a solution.

Next, I have proposed to use binary search to find the smallest α for which LP 2.1 is solvable [Dissertation, Alg 4.2]. This algorithm has an input (ϵ_U) that describes how close we want to get to the optimal error. Providing it is necessary due to the consequences of Thesis 2.4. Nevertheless, this way we can approximate the optimal solution arbitrarily close.

I have proven the polynomial complexity of this method. The underlying linear program contains no integer variables, hence it can be solved in polynomial time. This LP is run $\log_2(1/(\epsilon_U \min_{i,j} \gamma_{ij}))$ times, meaning that the whole method is indeed polynomial.

In this thesis I have given an iterative algorithm for the VRA-1N-mD-Unlimited problem, which quickly converges to an optimal setting. This is a notable result, as according to Thesis 2.5 no significantly better solution can be given.

[6] i.e., having a solution that can be written in a finite form using integer constants and the $+$, $-$, $\cdot$, $/$ and the nth root ($n \in \mathbb{Z}^+$) operators only

[7] In this LP I have provisionally relaxed the $\sum f_j = 1$ constraint and introduced variables $\hat{f}_j$ to avoid confusion. The reason is that I wanted to enforce $f_j > 0$, meaning that a link cannot be disabled, and this was the easiest way.

LP 2.1 VRA-1N-mD-Unlimited, Given α

$$
\begin{aligned}
\textit{indices:}\quad & i = 1 \dots D \\
& j = 1 \dots k \\
\textit{constants:}\quad & \alpha \quad (\alpha \geq 1,\ \ \alpha \in \mathbb{R}) \\
& \gamma_{ij} \quad (\gamma_{ij} \in \mathbb{Q},\ \ \gamma_{ij} \geq 0,\ \ \forall i : \sum_{n=1}^{k} \gamma_{in} = 1) \\
& \sigma_{ij} \quad = \operatorname{sgn}(\gamma_{ij}) \\
\textit{variables:}\quad & \hat{f}_j \quad (\hat{f}_j \geq 1,\ \ \hat{f}_j \in \mathbb{R}) \\
\textit{objective:}\quad & \text{minimize} \sum_{j=1}^{k} \hat{f}_j \\
\textit{constraints:}\quad & 0 \leq \sum_{n=1}^{k} \sigma_{in} \hat{f}_n - \frac{\hat{f}_j}{\gamma_{ij}\alpha}, \quad \forall i, j : \gamma_{ij} > 0
\end{aligned}
$$

4.2.3 Limited Number of Links

Thesis 2.7. *I have given two different, Integer Linear Program-based optimal solutions to the VRA-1N-mD problem. I have also given a pseudo-polynomial running time heuristic to the same problem.*

In this limited number of parallel links version of VRA-1N-mD there always exist at least one optimal solution, as there are only finite number of possible link allocations with at least one being the best. Clearly all link allocations are valid for which $e_j > 0\ \forall j$ and $\sum_{j=1}^{k} e_j \leq Q$. Also, as shown at the beginning of Sec. 4.2.1, there could be valid allocations for which $e_j > 0\ \forall j$ and $E_i \leq Q\ \forall i$ hold, but $\sum_j e_j \leq Q$ does not hold. This means that there are at least $\binom{Q}{k}$ valid link allocations (see the explanation at Thesis 1.3), which calls for a more efficient solution than the simple exhaustive search.

Consequently, I have given an Integer Linear Program that finds an optimal solution to the VRA-1N-mD problem [Dissertation, LP 4.4]. This ILP, however, operates with a large number of auxiliary integer variables increasing the running time, hence I have proposed a faster solution, too.

This second method is an ILP-based iterative solution. First, I have modified LP 2.1 for the limited number of links problem variant: I have added the constant Q, changed the real variables f_j to the positive integer variables e_j and finally added a new constraint: $\sum_{j=1}^{k} \sigma_{ij} e_j \leq Q\ (\forall i)$ [Dissertation, LP 4.3]. Using this ILP in a binary search framework [Dissertation, Alg. 4.4] results in an arbitrarily good approximation for the VRA-1N-mD problem. Due to the finite number of possible allocations, however, it is possible to give an absolute lower bound on the difference of the errors for two link allocation settings: I have proven that $\Delta U \geq (Q \max_{i,j} g_{ij})^{-2}$ [Dissertation, Lemma 21]. Using this constant in the stop condition of the binary search will lead to an *optimal solution*.

This latter iterative solution turned out to be considerably faster in practice than the first, monolithic ILP. Yet, it is still based on an ILP, therefore the polynomial running time

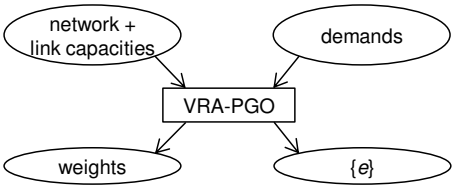


Figure 7: Virtual Resource Allocation–Peer-Global Optimization

is not guaranteed. Consequently, I have proposed a third, heuristic approach [Dissertation, Alg. 4.5]. The idea is to somehow represent matrix G of a VRA-1N-mD problem with a vector of length k, and treat that as vector g of a VRA-1N-1D problem. The latter can be solved quickly, as described in the previous subsection. What remains is to find an efficient method for the matrix G to vector g mapping. Solving the VRA-1N-mD-Unlimited problem does exactly this: the resulting f_js can simply be treated as vector g. Certainly some information is lost during the matrix to vector conversion, which can lead to suboptimal results. That is, nevertheless, acceptable as this method is a heuristic only.

4.3 Peer-Global Optimization

There is a fundamental problem with the Peer-Local Optimization approach: as the errors of the local optimizations propagate downstream with the data flows they encounter other imperfect splitting points, whereby local errors can enlarge or weaken each other's effect. The unintended effects of this kind of cascade errors can be avoided by minimizing the errors concurrently, in a centralized manner. This is what I call *Peer-Global Optimization.*

Thesis Group 3. [J1] *I have proposed and comprehensively studied another solution for OSPF Traffic Engineering, which I call Virtual Resource Allocation–Peer-Global Optimization. I have proven that it is NP-complete and cannot be approximated efficiently. I have also given an Integer Linear Program that finds an optimal solution and observed that Peer-Local Optimization can be used here as a faster heuristic.*

The optimization task here is the same as for Peer-Local Optimization: given a capacitated network and a set of demands determine for each link a weight and the number of parallel links that together minimize the maximal link utilization (see Fig. 7). At this approach, however, we solve the problem concurrently for all the nodes in the network so that we can reach a theoretically optimal virtual resource allocation.

4.3.1 Definition

Thesis 3.1. *I have identified and formally described the Peer-Global Optimization problem, which can provide a near-optimal solution for OSPF Traffic Engineering.*

Before the formal definition first let us see the notations for Virtual Resource Allocation–Peer-Global Optimization (VRA-PGO), summarized in Table 3. We are given a directed graph (V, F) representing a *network*, with *capacities* c_l for each link l and a *set of demands*,

Notation	Description
V	set of vertices (nodes) in the network
F	set of edges (physical links) in the network
S_n	set of (physical) links originating at node $n \in V$
$\|S_n\|$	number of (physical) links originating at node n
$c_l \in \mathbb{Q}^+$	capacity of link $l \in F$
$w_l \in \mathbb{Q}^+$	weight of link l
$h_l \geq 0$ ($h_l \in \mathbb{Q}$)	total actual traffic volume on link l
$e_l > 0$ ($e_l \in \mathbb{Z}$)	number of parallel links (both physical and virtual) at the place of link l
$E_n = \sum_{l \in S_n} e_l$	number of (physical and virtual) outgoing links at node n
D	number of demands
$O_d \in F$	originating node of demand d ($1 \leq d \leq D$)
$D_d \in F$	destination node of demand d
$G_d \in \mathbb{Q}^+$	traffic volume for demand d
$R \geq 0$ ($R \in \mathbb{Z}$)	maximal number of virtual links per node
$\beta \in \mathbb{Q}^+$	maximal link utilization

Table 3: VRA-PGO notation summary

each given by its originating and destination nodes, and the offered traffic volume: $\{O_d, D_d, G_d\}_{d=1}^{D}$. The *maximal number of virtual links* that can be applied at a node (R) is given as well. We are looking for *link weights* w_l and *parallel link number* e_l (including the physical and virtual links) for each link l, which minimizes the *maximal link utilization* $\beta = \max_{l \in F} h_l/c_l$, such that $E_n \leq |S_n| + R$ ($\forall n \in V$).

In the previous two subsections it was more convenient to limit the total number of links (both physical and virtual), requiring $E \leq Q$ and $E_i \leq Q$. With regard to the numerical evaluation, however, limiting the number of virtual links is more practical, as different nodes can have different number of outgoing physical links. For this reason throughout this subsection, unless stated otherwise, I limit the number of virtual links ($E_n - |S_n| \leq R$). Note, for Section 4.1 with a simple $R = Q - k$ substitution we can have a limit on the number of virtual links ($E - k \leq R$), too. Likewise, most of my findings in Sect. 4.2 are about the unlimited links variant of the VRA-1N-mD problem, but the rest are also trivial to transform to the have a limit on the number of virtual links only.

The problem can now be formulated as follows.

Problem 7, VRA-PGO. Given (V, F), $\{c_l\}$, D, $\{O_d, D_d, G_d\}$, and R, find $\{w_l\}$ and $\{e_l\}$ that minimizes β, such that $E_n \leq |S_n| + R$ ($\forall n \in V$).

4.3.2 Optimal Solution

Thesis 3.2. *I have given an Integer Linear Program that finds an optimal solution to the Peer-Global Optimization problem.*

This ILP is presented at [Dissertation, LP 5.1]. It is fairly lengthy: it contains 13 sets of constants, 8 sets of variables and 15 sets of constraints. I have also given recommendations for the values of the constants, which are introduced in this ILP, to be most suitable for practical computations.

The importance of this ILP is that it gives a theoretical lower bound on the error. On the other hand, as it contains lots of integer-valued auxiliary variables, for larger networks it is very slow to solve in practice. Nevertheless, I was able to solve it for several practical example scenarios to gain reference values for the performance evaluation of the heuristics (see Sec. 4.3.4).

In this thesis I have given a slow, but optimal solution to the Peer-Global Optimization problem. Note that the algorithms given for Peer-Local Optimization can be considered as quicker, but suboptimal heuristics for the same global problem. In the following subsection I show that finding a fast and even near-optimal solution is impossible as the problem is computationally hard by its nature.

4.3.3 Computational Complexity

Thesis 3.3. *I have proven the NP-completeness of the Peer-Global Optimization problem and several of its variants.*

For the NP-completeness proof first I have slightly reformulated the Peer-Global Optimization problem to be a decision problem. Furthermore, in this first approach I have taken the link weights as input parameters.

Problem 8, Virtual Resource Allocation–Peer-Global Optimization with Given Weights (VRA-PGO-GW).

INSTANCE. A directed graph (V, F) representing a *network* with *capacities* $c_l \in \mathbb{Q}^+$ and *link weights* $w_l \in \mathbb{Q}^+$ for each link $l \in F$. A *set of demands* $\{O_d \in F, D_d \in F, G_d \in \mathbb{Q}^+\}_{d=1}^{D}$. The *maximal number of virtual links* that can be applied at a node: $R \in \mathbb{Z}$ $(R \geq 0)$. The *maximal link utilization*: $\beta \in \mathbb{Q}^+$.

QUESTION. Is there a VRA assigning $e_l > 0$ $(e_l \in \mathbb{Z})$ number of links to each physical link $l \in F$, such that $E_n \leq |S_n| + R$ $(\forall n \in V)$ and $\max_l h_l/c_l \leq \beta$?

In the Question above $|S_n|$ can be calculated from (V, F); E_n and h_l can be calculated from (V, F), $\{e_l\}_{l\in F}$, $\{w_l\}_{l\in F}$ and from the set of demands. The only non-trivial point is the calculation of h_l, but it also can be done in polynomial time [Dissertation, Alg. 5.1].

This definition can be changed in several ways to obtain different versions of the problem, including the following:

- VRA-PGO: In this case setting the link weights, and this way defining the routing of the demands, is part of the problem, too. This variant is similar to VRA-PGO-GW, only the link weights are moved from the Instance to the Question.
- VRA-PGO-GW-SD (SINGLE DEMAND): In this variant we have only one origin–destination–traffic volume triplet $(D = 1)$.
- VRA-PGO-GW-Q: The definition of VRA-PGO-GW contains a bound on the number of virtual links $(E_n - |S_n| \leq R)$. In this version a limit on the total number of links is used instead $(E_n \leq Q)$.

- VRA-PGO-GW-ABS (Absolute Error): Here instead of the relative error (utilization) β, we have an absolute error, δ, requiring $\max_l(h_l - c_l) \leq \delta$.

By combining the definitions given above, several other equally valid variants of the VRA-PGO problem could be created. Fortunately, my proofs about computational complexity can be generalized relatively easily to many of these new cases.

I have proven that VRA-PGO-GW is NP-complete [Dissertation, Theorem 22]. In the proof first I have shown that the problem is in NP, then that it is NP-hard. The latter part is the harder one, where I have used Karp-reduction to reduce the 3SAT problem to VRA-PGO-GW. I have also modified this proof to prove the NP-completeness of VRA-PGO-GW-Q and VRA-PGO-GW-ABS as well. In Appendix C.3.1 of my dissertation I have presented an alternative proof of [Theorem 22], which I have also modified to prove that VRA-PGO is NP-complete, too.

Finally I have proven that VRA-PGO-GW-SD is also NP-complete [Theorem 26]. This statement is important, as it will form the basis of the following two theses. The proof itself is an extension of the first proof of [Theorem 22] and so it is also based on a 3SAT to VRA-PGO-GW-SD reduction.

Thesis 3.4. *I have formulated two variants of Peer-Global Optimization as NP optimization problems and have proven that it is impossible to computationally efficiently approximate their optimal solution within* every *constant ratio (unless $P = NP$).*

To examine the approximability of a problem the first step is to formulate it as an NP optimization (NPO) problem [20]:

Problem 9, Minimal Error Virtual Resource Allocation–Peer-Global Optimization with Given Weights (MIN-VRA-PGO-GW, shortly MVPG).

Instance. A directed graph (V, F) representing a *network* with *capacities* $c_l \in \mathbb{Q}^+$ and *link weights* $w_l \in \mathbb{Q}^+$ for each link $l \in F$. A *set of demands* $\{O_d \in F, D_d \in F, G_d \in \mathbb{Q}^+\}_{d=1}^{D}$. The *maximal number of virtual links* that can be applied at a node: $R \in \mathbb{Z}^+$.

Solution. A virtual resource allocation assigning $e_l > 0$ ($e_l \in \mathbb{Z}$) number of links to each physical link $l \in F$, such that $E_n \leq |S_n| + R$ ($\forall n \in V$).

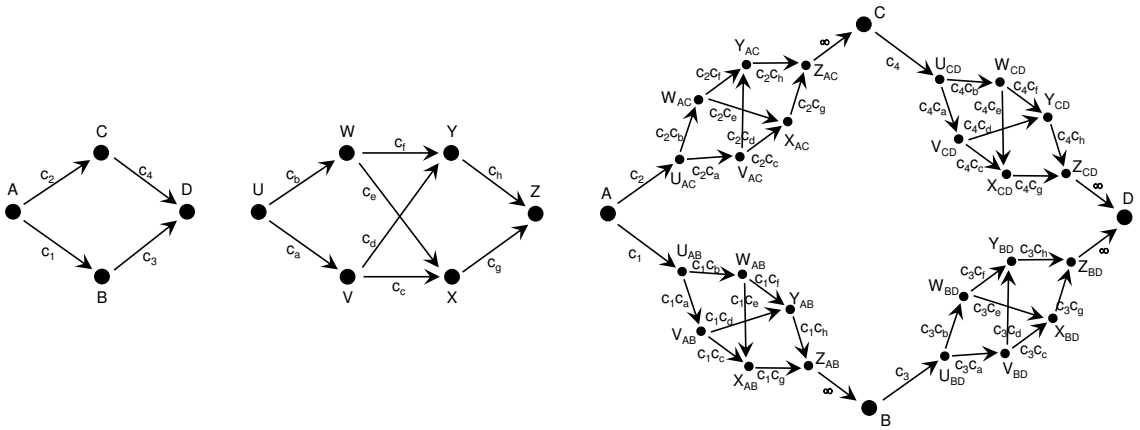
Measure. The *maximal link utilization* $\beta = \max_l h_l/c_l$.

Goal. Minimize the measure.

Problem 10, Minimal Error VRA-PGO with Given Weights for a Single Demand (MIN-VRA-PGO-GW-SD, shortly MVPGS).

This is essentially the same as MVPG, but has only exactly one demand.

First, I have checked that both of these problems conform to the NPO requirements [20]. Next, I have proven that no polynomial time algorithm exists that approximates the optimum of MVPG better than a factor of 6/5 [Dissertation, Theorem 27]. The proof is an extension of the corresponding NP-completeness proof of VRA-PGO-GW [Thm. 22]. Similarly, I have also proven that no polynomial time algorithm exists that approximates the optimum of MVPGS better than a factor of 18/17 [Thm. 28]. The proof relies on the NP-completeness proof of VRA-PGO-GW-SD [Thm. 26] and on the proof of inapproximability of MVPG [Thm. 27].



(a) I_A (b) I_B (c) $I_A \otimes I_B$

Figure 8: MVPGS compounding

From these statements it follows that it is impossible to computationally efficiently approximate the optimal solutions of these problems within *every* constant ratio (of course unless P = NP). In other words this means that *MVPG and MVPGS are not part of the Polynomial Time Approximation Scheme (PTAS) class.*

Thesis 3.5. *I have shown that the optimal solution of the MVPGS problem (which is a variant of Peer-Global Optimization) cannot be approximated with a polynomial time algorithm within* any *constant ratio (unless P = NP).*

This thesis corresponds to [Dissertation, Theorem 29]. The proof applies the "inapproximability gap amplification technique", which has recently been introduced in [13] to prove a similar inapproximability for the OSPF ECMP link weight configuration problem.

I have first introduced the $\otimes$ (compound) operator for MVPGS instances. From two instances I_A and I_B a new instance $I = I_A \otimes I_B$ can be crated by compounding if both of the following conditions hold:

1. the traffic volume of the demand to be transmitted in I_B is 1,
2. the allowed maximum number of virtual links (R) is identical in I_A and I_B.

An example of compounding is shown in Fig. 8. The capacities are shown next to the links and each link weight is one unit. In I_A the demand is $A \to D : 1$, in I_B it is $U \to Z : 1$. $R = 1$ in all these instances.

Furthermore, for an MVPGS instance I I have introduced $OPT(I)$ denoting the measure for the optimal solution, i.e., the minimal β. I have also introduced the following notation:

$$I_0 = \otimes^0 I = I$$

$$I_1 = \otimes^1 I = I \otimes I$$

$$\vdots$$

$$I_k = \otimes^k I = I \otimes (\otimes^{k-1} I)$$

Next I have proven by induction the following [Dissertation, Lemma 30]:

Lemma. *Let I be an instance of MVPGS with $OPT(I) \geq 1$. Then for any $k \in \mathbb{Z}$, $k \geq 0$: $OPT(\otimes^k I) = (OPT(I))^{k+1}$.*

Based on this lemma and [Theorems 26 and 28] I have proven [Theorem 29], which is identical to the claim of this thesis. In other words this means that *MVPGS is not part of the APX class.*

4.3.4 Numerical Evaluation

The previous complexity-related results state the hardness of Peer-Global Optimization in general. They, however, do not necessarily mean that in practical networks no effective solution can exist. To see how the different algorithms perform in realistic environments I have implemented a simulation framework. The simulator takes a capacitated network and a set of demands as inputs and solves the Virtual Resource Allocation problem using several different algorithms. I have implemented the framework and the optimization algorithms in C++ using the powerful LEMON Graph Library [16]. I have solved the embedded linear programs using the IBM ILOG CPLEX Optimizer [17].

I have included the following seven optimization approaches in my simulator:

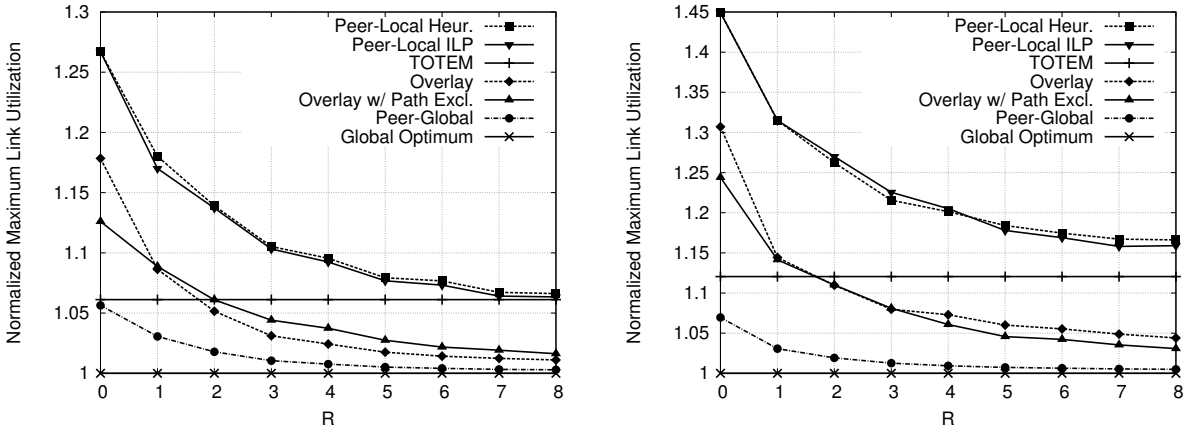
1. Overlay Optimization, as described in Sec. 4.1.
2. Overlay Optimization with Path Exclusion. This is a slight modification of Overlay Optimization that allows path disabling (i.e. $e_i = 0$ is permitted).
3. Peer-Local Optimization with the ILP-based iterative solution described in Sec. 4.2.3.
4. Heuristic Peer-Local Optimization, as presented at the end of Section 4.2.3.
5. Peer-Global Optimization, outlined in Sec. 4.3.2.
6. Global Optimization as described below.
7. OSPF Weight Optimization (or OSPF-TE), see below.

In the last case the goal is to set the link weights so that running stock OSPF with ECMP on top of this network will generate the minimal MLU (Maximal Link Utilization). In this case we are not applying virtual resources at all. This problem is proven to be NP-hard [5], but in the same paper a heuristic is proposed, which have been implemented in an open source toolbox, called TOTEM (TOolbox for Traffic Engineering Methods [14, 15, 21]). I have included it in my simulations to serve as a best-practice solution of OSPF Weight Optimization.

About *Global Optimization.* Taking the capacitated network and the demands and solving the related multi-commodity flow LP results in the optimal per link per demand traffic. If, by using an adequately sophisticated TE mechanism, the demands could be routed perfectly according to the solution of this LP then the theoretical minimal MLU could be reached. Accordingly, I have included a simple algorithm in my simulation platform that treats the outputs of this multi-commodity LP as actual traffic values. These results will then serve as reference values, since no algorithm (neither Peer-, nor Overlay-based) can perform better than this one. Furthermore, I will actually divide the MLU's of the different algorithms by this optimal MLU to have a normalized value, which is independent of the actual link bandwidths and traffic volumes. The result of this optimization will be denoted

Network	No. of nodes	No. of unidirectional links	Link capacities [units]
Abilene	12	30	100
Pan-European	16	46	100

Table 4: Network characteristics



(a) Abilene topology

(b) Pan-European topology

Figure 9: Simulation results

in the charts as *Global Optimum*. Certainly, when displaying MLU values, Global Optimum will be constant 1.0 due to the normalization.

In my dissertation I have presented my simulation results in detail and here I only highlight the most interesting ones. I have acquired these selected findings simulating two realistic network topologies: the first one is the well known North American Abilene network, the second is a simplified Pan-European optical core network (see Table 4). In both cases uniform link bandwidths of 100 units have been used. I had 5 demands in each simulation session and in each case the source and destination nodes were selected randomly. The traffic volumes has also been picked at random for each demand with uniform distribution on the $[5, 30]$ units interval. The maximal number of virtual links or paths (R) was varied between 0 and 8, inclusive. I have run the simulations 300 times (with all the seven algorithms) on a high-performance computer to decrease the variance of the results.

The most interesting results are shown in Fig. 9. TOTEM performed almost as good as Peer-Global Optimization for the no virtual link case, which is its theoretical lower bound. For $R = 0$ Peer-Local and Overlay Optimizations performed clearly worse than TOTEM, which is no surprise: running a VRA algorithm without virtual resources does not make much sense. However, allowing only two or three virtual links per node Overlay Optimization clearly overperformed TOTEM, getting as close as a few percents to the Global Optimum. The Peer-Local ILP and Peer-Local Heuristic algorithms performed the worst, but these are quick heuristics only for the VRA-PGO problem. The Peer-Global Optimization's MLU is well below TOTEM's even for $R = 1$ and it keeps decreasing as R increases, almost reaching the Global Optimum for only $R = 4$. This shows that Peer-Global Optimization does have

a high potential, but the currently applied heuristics are not taking full advantage of this, leaving space for future research for better ones.

Regarding the *resource consumption* of the different algorithms, I have observed very short ($\approx$ second, sub-second) running times for Overlay, Peer-Local and Global Optimizations, suggesting that these algorithms are likely to be suitable when short response times are needed, like real-time TE optimization. For TOTEM, the calculations took several tens of seconds, even exceeding a minute, which can still be practical for non-realtime tasks. The memory usage was modest, only 4–10 MB in these cases. However, with Peer-Global Optimization the average running times increased up to several hours. In this case the variance was much higher as well: the running times for a single session ranged from a couple of seconds to several days. The memory usage also varied from a few MB to almost 30 GB. This means that the proposed Peer-Global Optimization algorithm may not be a viable option in many of the practical cases.

5 Applicability of New Results

In my dissertation I have proposed and thoroughly examined the concept of Virtual Resource Allocation. Throughout the body of my work I have used OSPF Traffic Engineering as a descriptive use case. Certainly the proposed optimization frameworks and algorithms can directly be used in real communication networks applying OSPF TE to provide significantly better network performance.

Another possible field of application is the improvement of the WCMP protocol. Sections 3.2.1–3.2.3 of paper [2] introducing WCMP propose only heuristic solutions for rule table optimization. My algorithms listed in Thesis 1.3 could directly be applied to the WCMP problems as well and they always provide optimal solutions, not just approximations. Sec. 3.4.1 of my dissertation is devoted to the application of VRA for WCMP.

To show another use case, in the Fibbing proposal [8] if a routing has a single shortest path for a destination and two parallel paths are to be used with equal traffic share, advertisements of a fake node and a fake link has to be injected to the network. Likewise, if for example 33%–67% traffic ratio is to be achieved in an unequal load sharing case then two fake nodes and links have to be advertised. This shows that My VRA framework could be used with Fibbing to find the best approximation of an arbitrary split ratio using bounded number of fake entities.

COYOTE [22] is recently proposed TE scheme that allows traffic demand volumes to have some uncertainty. It applies the idea of Fibbing combined with some of my algorithms proposed for VRA-1N-1D to approximate optimal TE in their examined scenario.

These examples show that VRA can be applied in several contexts within in a network, where optimal resource pooling are to be achieved over legacy equal-split schemes.

Finally, I have prepared a patent application with my co-author Attila Kőrösi about the VRA concept. Ericsson, the multinational telecommunication equipment vendor company has found it worthwhile to purchase it and file the application with us as Inventors and Ericsson as Applicant [O1]. This also emphasizes the importance and utility of my work described here.

References

[1] D. Wischik, M. Handley, and M. B. Braun, "The resource pooling principle," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 47–52, 2008.

[2] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat, "WCMP: Weighted cost multipathing for improved fairness in data centers," in *Proceedings of the Ninth European Conference on Computer Systems*, ser. EuroSys, Apr. 2014, pp. 5:1–5:14.

[3] J. T. Moy, "OSPF Version 2," RFC 2328, Mar. 2013.

[4] International Organization for Standardization, "Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)," ISO/IEC 10589:2002, November 2002.

[5] B. Fortz and M. Thorup, "Increasing internet capacity using local search," *Computational Optimization and Applications*, vol. 29, pp. 13–48, 2004.

[6] N. Kang, M. Ghobadi, J. Reumann, A. Shraer, and J. Rexford, "Efficient traffic splitting on commodity switches," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '15, Dec. 2015, pp. 6:1–6:13.

[7] A. Maghbouleh, "Metric-based traffic engineering: Panacea or snake oil? A real-world study," in *27th North American Network Operators Group Meeting (NANOG27)*, Feb 2003.

[8] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford, "Central Control Over Distributed Routing," in *ACM SIGCOMM*, August 2015, pp. 43–56.

[9] O. Tilmans, S. Vissicchio, L. Vanbever, and J. Rexford, "Fibbing in action: On-demand load-balancing for better video delivery," in *ACM SIGCOMM*, 2016, pp. 619–620.

[10] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering," RFC 3272, May 2002.

[11] Y. Wang and Z. Wang, "Explicit routing algorithms for internet traffic engineering," in *Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, 1999, pp. 582–588.

[12] Z. Wang, Y. Wang, and L. Zhang, "Internet traffic engineering without full-mesh overlaying," in *IEEE INFOCOM 2001*, vol. 1, April 2001, pp. 565–571.

[13] M. Chiesa, G. Kindler, and M. Schapira, "Traffic Engineering with Equal-Cost-MultiPath: An Algorithmic Perspective," in *IEEE INFOCOM 2014*, April 2014, pp. 1590–1598.

[14] G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescapè, B. Quoitin, S. P. Romano, E. Salvadori, F. Skivée, H. T. Tran, S. Uhlig, and H. Ümit, "An open source traffic engineering toolbox," *Comput. Commun.*, vol. 29, no. 5, pp. 593–610, Mar. 2006.

[15] H. Ümit, "Interior Gateway Protocol Weight Optimization Tool," Feb. 2007, retrieved Nov. 2018. [Online]. Available: http://www.poms.ucl.ac.be/totem/

[16] "LEMON Graph Library – Library for Efficient Modeling and Optimization in Networks," July 2014, Version 1.3.1. Retrieved Nov. 2018. [Online]. Available: http://lemon.cs.elte.hu/

[17] "IBM ILOG CPLEX Optimizer," Retrieved Nov. 2018. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer

[18] "Maple (mathematical software)," 2018, retrieved Nov. 2018. [Online]. Available: http://www.maplesoft.com/products/Maple/

[19] R. Joseph, *Galois Theory*, ser. Universitext. Springer New York, 1998, second edition.

[20] V. Kann, "On the Approximability of NP-complete Optimization Problems," Ph.D. dissertation, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, May 1992.

[21] "TOTEM Project: TOolbox for Traffic Engineering Methods," Latest version (3.2.1) released at Nov. 2008. Retrieved Nov. 2018. [Online]. Available: http://totem.run.montefiore.ulg.ac.be/

[22] M. Chiesa, G. Rétvári, and M. Schapira, "Lying your way to better traffic engineering," in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '16, 2016, pp. 391–398.

Publications

International journals

[J1] **Krisztián Németh**, Attila Kőrösi, Gábor Rétvári. "Optimal Resource Pooling over Legacy Equal-Split Load Balancing Schemes", *Computer Networks*, 127, Nov. 2017, pp. 243-265. (6/2=3 points)

[J2] Nikolett Bereczky, Amalia Duch, **Krisztián Németh**, Salvador Roura. "Quad-kd trees: A general framework for kd trees and quad trees", *Theoretical Computer Science*, 616, Feb. 2016, pp. 126-140. (6/4=1.5 points)

[J3] Péter Füzesi, **Krisztián Németh**, Niklas Borg, Rikard Holmberg, István Cselényi. "Provisioning of QoS enabled inter-domain services", *Computer Communications*, 26 (10), Jun. 2003, pp. 1070-1082. (6/5=1.2 points)

[J4] Gábor Fehér, **Krisztián Németh**, István Cselényi. "Performance Evaluation Framework for IP Resource Reservation Signaling", *Performance Evaluation*, 48 (1-4), May 2002, pp. 131-156. (6/3=2 points)

Hungarian journals

[J5] **Németh Krisztián**. "Hívásengedélyezés garantált minőségű hálózatokban (áttekintés)", *Híradástechnika*, ISSN: 0018-2028, LVII, 2002/9, pp. 2-4. (Reviewed, 2 points)

[J6] Cselényi István, Füzesi Péter, **Németh Krisztián**. "Az internet szolgálatminőség fejlődése", *Magyar Távközlés*, ISSN: 0865-9648, 2000/2, pp. 26-31. (Not reviewed, 1/3=0.333 points)

[J7] **Krisztián Németh**. "IP Multicasting over ATM", *Magyar Távközlés, Selected Papers*, ISSN 0865-9648, 1999, pp. 11-15. (In English, not reviewed, 2 points)

[J8] **Németh Krisztián**. "IP multicast ATM felett", *Magyar Távközlés*, ISSN: 0865-9648, 1998/7, pp. 3-6. (Not reviewed, [J7] is a revised translation of [J8], 0 points)

International conferences

[C1] **Krisztián Németh**, Attila Kőrösi, Gábor Rétvári. "Enriching the poor man's traffic engineering: Virtual link provisioning for optimal OSPF TE", *Networks 2014*, Funchal, Madeira Island, Portugal, September 2014. (3/2=1.5 points)

[C2] Nikolett Bereczky, Amalia Duch, **Krisztián Németh**, Salvador Roura. "Quad-K-d Trees" *Latin American Theoretical INformatics* (LATIN 2014), Montevideo, Uruguay, March 31 - April 4, 2014, Proc. LNCS 8392, pp. 743-754. (3/4=0.75 points)

[C3] **Krisztián Németh**, Attila Kőrösi, Gábor Rétvári. "Optimal OSPF Traffic Engineering using Legacy Equal Cost Multipath Load Balancing", *IFIP Networking 2013*, Brooklyn, New York, USA, May 2013 (3/2=1.5 points)

[C4] Tibor Cinkler, Réka Kosznai, Péter Balázs Soproni, **Krisztián Németh**. "GSP, the Generalised Shared Protection" *9th International Conference on Design of Reliable Communication Networks* (DRCN 2013), Budapest, Hungary, March 2013. (3/4=0.75 points)

[C5] **Krisztián Németh**, Gábor Rétvári. "Traffic Splitting Algorithms in Multipath Networks: Is the Present Practice Good Enough?", *Networks 2012*, Rome, Italy, October 2012. (3 points)

[C6] Zsolt Kovácsházi, Gábor Papp, **Krisztián Németh**. "A Hybrid Multicast Video Distribution Method: a Technology for E-Entertainment", *2005 Networking and Electronic Commerce Research Conference* (NAEC 2005), Riva del Garda, Italy, October 2005, pp. 1-11. (3/3=1 points)

[C7] Niklas Borg, Rikard Holmberg, Péter Füzesi, **Krisztián Németh**. "NAIS – Network Architecture for Inter-Domain Services", *Networks 2002*, Munich, Germany, June 2002. Proceedings ISBN: 3-8007-2711-0 (3/4=0.75 points)

[C8] **Krisztián Németh**, Péter Füzesi. "An Analysis of IP Resource Reservation Protocols", *Networks 2002*, Munich, Germany, June 2002. Proceedings ISBN: 3-8007-2711-0, pp. 213-220 (3/2=1.5 points)

[C9] István Moldován, **Krisztián Németh**. "Quality of Service Architectures Using MPLS Networks", *9th IFIP Working Conference on Performance Modelling and Evaluation of ATM and IP Networks*, Budapest, Hungary, June 2001. Proceedings ISBN 963 420 694 8 (3/2=1.5 points)

[C10] István Moldován, **Krisztián Németh**, Tibor Cinkler. "Merging in MPLS Networks", *IEEE ICT 2001*, Bucharest, Romania, June 2001 (3/3=1 points)

[C11] Csaba Simon, **Krisztián Németh**, Sándor Székely. "Point-to-Multipoint ATM Signalling Performance Measurements", *8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, UK, July 2000. Proceedings: Networks UK, D. D. Kouvatsos (Ed.) ISBN 0-9540151-1-8 (3/3=1 points)

[C12] **Krisztián Németh**, Gábor Fehér, István Cselényi. "Simulation Study for IP Resource Reservation", *8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, UK, July 2000. Proceedings: Networks UK, D. D. Kouvatsos (Ed.) ISBN 0-9540151-1-8 (3/3=1 points)

[C13] Gábor Fehér, **Krisztián Németh**, István Cselényi. "Router Benchmarking Framework for QoS Signaling", *8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, Ilkley, UK, July 2000. Proceedings: Networks UK, D. D. Kouvatsos (Ed.) ISBN 0-9540151-1-8 (3/3=1 points)

[C14] István Cselényi, Gábor Fehér, **Krisztián Németh**. "Benchmarking of Signaling Based Resource Reservation in the Internet", *Networking 2000*, IFIP-TC6, Paris, France, May 2000. Proceedings: LNCS 1815; Networking 2000: Broadband Communications, High Performance Networking and Performance of Communication Networks; ed. Guy Pujolle et al., ISSN 0302-9743, ISBN 3-540-67506-X (3/3=1 points)

[C15] **Krisztián Németh**, Krzysztof Szarkowicz. "IP Multicasting over ATM", *7th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks*, IFIP WG 6.3, 6.4, Antwerp, Belgium, June 28-30, 1999 (3/2=1.5 points)

[C16] Gábor Fehér, **Krisztián Németh**, Markosz Maliosz, István Cselényi, Joakim Bergkvist, David Ahlard, Tomas Engborg. "Boomerang – A Simple Protocol for Resource Reservation in IP Networks", *IEEE Workshop on QoS Support for Real Time Internet Applications in conjunction with Real-Time Technology and Applications Symposium* (RTAS), Vancouver, Canada, June 2-4, 1999 (3/7=0.429 points)

Hungarian conferences

[C17] **Németh Krisztián**. "Hívásengedélyezés garantált minőségű csomagkapcsolt hálózatokban", *PKI Tudományos Napok 111*, Budapest, November 2002. (1 points)

Other publications

[O1] Attila Kőrösi, **Krisztián Németh**. "Methods and packet network devices for forwarding packet data traffic", *International Patent Application*, Int. Application No.: PCT/EP2013/060404, International Filing Date: 21 May, 2013, Publication No.: WO/2014/187475, Publication Date: 27 Nov., 2014 (2/2=1 points)

[O2] Gábor Fehér, **Krisztián Németh**, András Korn, István Cselényi. "Benchmarking Terminology for Resource Reservation Capable Routers", *IETF RFC* 4883, July 2007 (0 points)

Total publication score: 33.212 points