Computer Networks 127 (2017) 243–265

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Optimal resource pooling over legacy equal-split load balancing schemes

Krisztián Németh [a,*], Attila Kőrösi [a,b], Gábor Rétvári [a,b]

[a] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudósok krt. 2, Budapest 1117, Hungary
[b] MTA-BME Information Systems Research Group, Magyar tudósok krt. 2, Budapest 1117, Hungary

ARTICLE INFO

Article history:
Received 18 March 2017
Revised 14 July 2017
Accepted 19 August 2017
Available online 31 August 2017

Keywords:
Resource pooling
Load balancing
Traffic splitting
Non-equal splitting
ECMP

ABSTRACT

Splitting traffic flows to different data paths is crucial in current and future networks. Traffic division serves as the basis for load balancing between application servers, optimal Traffic Engineering, using multiple paths in data centers, and several other places of an end-to-end connection. Unfortunately, by allowing only equal division amongst the parallel resources, existing technologies often cannot realize the optimal traffic splitting, which can have serious negative consequences on the network performance.

In this paper we present a flexible and effective traffic splitting method that is incrementally deployable and fully compatible with practically all existing protocols and data planes. Our proposal, called Virtual Resource Allocation (VRA), is based on setting up virtual resources alongside existing ones, thereby tricking the legacy equal traffic splitting technology into realizing the required non-equal traffic division over the physical media. We propose several VRA schemes, give theoretical bounds on their performance, and also show that the full-fledged VRA problem is NP-complete in general. Accordingly, we provide solution algorithms, including an optimal, but necessarily slow method and several quick heuristics. Our simulations show that VRA has huge practical potential as it allows approaching an ideal traffic split using only a very limited set of virtual resources. Based on the results, we also give detailed suggestions on which algorithm to apply in different scenarios.

1. Introduction

Unity is strength. Treating separate network resources as one and sharing it among the 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 [2], load balancing between application level servers (such as web-servers or database servers), load balancing in Traffic Engineering. Content Delivery Networks [3] are also a form of resource pooling, just as cloud storage and cloud computing [4]. To realize these services, data centers are being installed rapidly, also often utilizing equal-length parallel paths, which are, in many of the cases, asymmetric in capacity [5]. Furthermore, several new concepts, such as network virtualization and Software Defined Networking (SDN) [6] 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 service can be offered on the same infrastructure, due to 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 usually 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 [7] or IS-IS [8] Equal-Cost Multipath (ECMP) is used to distribute the traffic over the shortest paths with the same cost. ECMP, however, is only able to split traffic between these paths uniformly, even if they

* Corresponding author.
E-mail addresses: krisztian.nemeth@tmit.bme.hu (K. Németh), korosi@tmit.bme.hu (A. Kőrösi), retvari@tmit.bme.hu (G. Rétvári).

http://dx.doi.org/10.1016/j.comnet.2017.08.017
1389-1286/

have different capacities, which poses a giant barrier when aspiring to an optimal Traffic Engineering [5,9–11].

There are several existing proposals which target specific cases of this issue. Weighted Cost Multipathing (WCMP, [5]), for example, aims unequal traffic splitting at data centers. It assumes SDN-capable switches, and operates by replicating rule table entries. Niagara [10] 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 [12] is lately proposed, 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 [13]. These proposals, however, are more or less coupled to a single field of application, and, 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, we introduce a technique called Virtual Resource Allocation (VRA). The basic 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. We then group the virtual resources and assign them to the physical ones, thereby tricking the legacy equal splitting technology into realizing the required non-equal load division over the existing media.

As an example, we can install two virtual machines on the stronger web-server and present them, along with the unmodified weaker server, to the load balancer. It then sees three servers, and by realizing equal split between them the stronger server will eventually end up with 2/3 part of the total load, as desired. In a similar fashion, installing virtual links or paths alongside the physical ones (which, in practice, can be carried out via some administrative settings), ECMP's equal-split limitation can be amended. If, for example a $25 - 75\%$ traffic proportion is desired on two, equal cost shortest paths A and B, then by installing two virtual paths parallel to B, and presenting these four to ECMP, it will happily realize the expected traffic split rate.

The engineering problem to solve in VRA is then to come up with an optimal setting of virtual resources so that a predefined non-equal traffic split ratio is approximated sufficiently with limited resource usage. Furthermore, placing VRA in a broader scope, other, network-wide goals can be targeted as well.

Later on in this paper Traffic Engineering (TE) in IP networks will be used to introduce the VRA proposal. Let us emphasize, however, that TE is just a descriptive example application of the VRA concept, and its possible fields of usage are much broader. For instance, Fibbing can be enhanced by our algorithms proposed for virtual resource mapping. Software Defined Networking is a recent and promising trend in the IP world. While VRA does not depend on SDN, there already exist application possibilities for VRA within an SDN framework and more are likely to come. To name one, in data centers parallel shortest path are very frequent, but their capacities tend to be asymmetric [5], causing ECMP to be a suboptimal tool for splitting. WCMP, using OpenFlow, is designed to cope with this challenge. Yet, it can be enhanced by the algorithms described in this paper to minimize to forwarding table entries while keeping the oversubscription rate under a limit.

Main contributions. We present a precise *VRA problem definition* along with *theoretical error bounds* for different scenarios. We also provide *algorithms* to solve the problem *under different real-life constraints*. Our proposition 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 *highly compatible*, meaning that it does not necessitate installing any new hardware or software component in the network. Finally, VRA is *extremely efficient*, as our 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.

We *show the NP-completeness* of the full-fledged version of the VRA problem, and also show that *no polynomial time algorithm can approximate the optimal solution within* any*constant ratio*. Yet, *we propose an Integer Linear Program* (ILP) as an optimal solution along with *quick heuristics*.

Finally, we present our *simulation evaluation*, which includes our algorithms as well as the existing best-practice solution. Our results underpin that the *VRA approach has a huge practical potential*. This, together with the easy deployability make *VRA an ideal choice for network operators*.

Previous works. This paper is a continuation, and in some sense, a completion of an earlier work carried out by the same authors, which has been published in [14] and [15], and which address the Overlay Optimization an Peer-Local Optimization. The main findings of those conference papers are summarized in this one for the sake of completeness, but everything else in this paper, unless directly cited, are, to our best knowledge, first published here. These include, but not limited to the Peer-Global Optimization ILP, the results about the computational complexity of the problem, and the whole simulation evaluation. While our earlier efforts focused solely on TE, this time we have generalized it into the much more universal VRA context, which have certainly affected the structure of the whole paper.

Organization. The rest of this paper is structured as follows. Section 2 introduces the VRA concept with some simple examples. Sections 3 and 4 carry our main theoretical results about different versions of the original problem. A numerical evaluation is described in Section 5, followed by an overview of the most important related works in Section 6. Finally, Section 7 concludes this paper. Appendix A contains an ILP that solves the full-fledged VRA problem, and Appendix B reveals our theorems and proofs about its computational complexity.

2. Virtual resource allocation overview

In this section we overview the Virtual Resource Allocation concept, using Traffic Engineering as a descriptive example.

The idea behind VRA is fairly simple and is best explained by a small sample problem. Consider the triangular network that is shown, with the link capacities, 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 (also often called link costs, link metrics or SPF (Shortest Path First) metrics), thereby we could easily create two equal cost shortest paths (i.e. paths with minimal total 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 to split the traffic equally between the shortest paths, implying a 150% load on $A - B$ and $B - C$.

If, however, we could set up a virtual link on top of the existing $A - C$ link, and expose it to OSPF (see Fig. 1(c)), it would happily split the traffic in three, sending one third on the $A - B - C$ path and the rest on the $A - C$ physical link (Fig. 1(d)). Naturally, installing a virtual link over the $A - C$ physical link does not change its capacity, it only enables OSPF ECMP to use its full potential in this case. The link weights would also remain unchanged, and the new virtual link would have the same weight as the respective physical one (2 in our example). By this simple administrative intervention we can route the traffic through this network without exceeding the link capacities.

There are several possible ways to set up a virtual link parallel to an existing one. These options include Ethernet VLANs, IP-