ARTICLE IN PRESS

Pervasive and Mobile Computing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

Fast track article

Augmenting flows for the consistent migration of multi-commodity single-destination flows in SDNs☆

Sebastian Brandt, Klaus-Tycho Foerster*, Roger Wattenhofer

ETH Zürich, Gloriastrasse 35, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Software defined networks
Congestion
Multi-commodity flow
Anycast
Flow augmentation
Consistent migration

ABSTRACT

Updating network flows in a real-world setting is a nascent research area, especially with the recent rise of Software Defined Networks. While augmenting *s-t* flows of a single commodity is a well-understood concept, we study updating flows in a multi-commodity setting: Given a directed network with flows of different commodities, how can the capacity of some commodities be increased, without reducing capacities of other commodities, when moving flows in the network in an orchestrated order? To this extent, we show how the notion of augmenting flows can be efficiently extended to multiple commodities for applications with a single logical destination. We also show that our methods induce stronger consistency settings than previous work. Lastly, we prove the consistent migration to new demands to be NP-hard for unsplittable flows, and discuss extensions for the case of multiple source–destination pairs.

1. Introduction

The rise of *Software Defined Networks (SDNs)* has sparked an increasing interest in applying network flow algorithms. While smart traffic engineering is not only studied or enabled in SDNs, utilizing the available bandwidth almost completely is one of the central topics in SDN research [1]. The algorithmic tool to manage network traffic in an efficient way is provided by flow algorithms.

Since network traffic is highly dynamic, existing SDN solutions [1–5] frequently re-compute the optimal way to route traffic demands, usually using an approach based on linear programming (LP), often accepting the overhead that a new solution will re-route many existing flows that did not change their demands.

In this article we propose to abandon LP-based solutions in favor of path augmentation, a technique developed 60 years ago [6]: We believe that SDNs should be managed in an incremental way. If a commodity (a source–destination node pair) wants to reduce its bandwidth, no changes need to be made. If a commodity wants to increase its bandwidth (or a new commodity is introduced to the network), we try to increase its flow by using path augmentation. In the best case, the increased demand fits without changing any of the other flows. However it can be that re-routing (also called migration) of some other flows is necessary, conceivably even recursively.

The problem of flow migration still has multiple open research questions: It is not clear in general (1) how to classify when congestion-free migration is possible, (2) to what solution one should actually attempt to migrate, (3) how to reasonably bound the migration time.

☆ A preliminary extended abstract appeared in the Proceedings of the 17th International Conference on Distributed Computing and Networking (ICDCN '16), ACM, New York, NY, USA, http://dx.doi.org/10.1145/2833312.2833450, Brandt et al., (2016).

* Corresponding author.
E-mail addresses: brandts@ethz.ch (S. Brandt), foklaus@ethz.ch (K.-T. Foerster), wattenhofer@ethz.ch (R. Wattenhofer).

http://dx.doi.org/10.1016/j.pmcj.2016.09.012
1574-1192/

Moreover, there is another problem: Apart from a few exceptions that we discuss in the related work section, path augmentation was only developed for *s*-*t*-flow problems with a *single* commodity. In real networks we have *multiple* commodities, so we first need to generalize path augmentation to *flow augmentation*, a path augmentation technique supporting multiple commodities.

It turns out that generalizing path augmentation is not as easy as one may hope. As Hu notes in his influential article [7], "*it is unlikely that similar techniques can be developed for constructing multicommodity flows*".

This is why this article focuses on an important case of multi-commodity flow, the so-called *anycast* problem, cf. [8]. In the anycast problem, we have different commodities, one for each source node. All these commodities must be routed to an arbitrary set T of destination nodes (or from T to the source nodes). In contrast to general multi-commodity flow problems, it does not matter which commodity ends up at which destination, as long as the destination is in the set T. Commodities may route to *any* destination of the set T, hence anycast.

Using popular terminology, think of T as the set of servers of a cloud provider; customers do not care which server gets the (potentially enormous [9]) data, as long as "the data gets to the cloud". An analogous case can be made when, e.g., migrating virtual machines *inside* the cloud (i.e., anycast-based data center intra- [10] or inter-connections [11]). Furthermore, when data is requested from the cloud, the customer does not care either from which anycast-based [12] content delivery networks the (identical) data is obtained, cf. [13,14]. For example, streaming a video to digital media player, or a mobile phone requesting a web site currently under a distributed denial-of-service (DDoS) attack [15,16].

Applying our method of flow augmentation, we develop an efficient algorithm for consistently migrating to any desired feasible set of traffic demands. We require only one augmenting flow per commodity, minimizing network overhead. Thus, for the anycast setting, we solve all the three issues (1), (2), (3) mentioned above.

Furthermore, we show that our method can also be extended to stronger notions of consistency, by adding a polynomial number of intermediate updates to the flow migration. Lastly, we also prove that the consistent migration problem to new demands turns out to be NP-hard for unsplittable flows.

1.1. Structure of our article

After discussing the background and related work in Section 2, we continue with the model Section 3—where the term consistent migration is defined formally. In Section 4 we develop a technique to use augmenting flows for consistent migration in the anycast setting, before showing in Section 5 how to implement our method efficiently in practice. We also prove that our method works for stronger consistency models in Section 6, but that the corresponding problem for unsplittable flows is NP-hard, cf. Section 7. After discussing the case of general multi-commodity flows in Section 8, we conclude with Section 9.

2. Background and related work

To the best of our knowledge, the concept of flow augmentation has not yet been used in the context of consistent migration. Thus, we treat both topics separately, followed by a concise comparison of our work to current flow migration techniques.

2.1. Augmentation & multi-commodity flows

The notion of augmenting paths for single-commodity flows has been introduced in the seminal works of Ford and Fulkerson [6,17], with their concepts influencing thousands of publications to this day. In the last decades, there has been a great amount of research regarding (multi-commodity) flow problems. We refer to the textbooks by Cormen et al. [18] and Ahuja et al. [19] for an in-depth overview.

Hu [7] studied augmenting paths for a two-commodity setting and generalized the results of Ford and Fulkerson to maximize the simultaneous flow of two commodities. By limiting the problem to just two commodities, he introduced so-called backward and forward paths, which together allow for an augmentation of the network.

Furthermore, in 1978, shortly before the celebrated publication of the Ellipsoid method [20], Itai [21] published an improved version of Hu's two-commodity flow algorithm and showed that maximizing a two-commodity flow is as difficult as linear programming in the sense that they are polynomially equivalent.

However, while many further results were published for multi-commodity flow problems in general and augmenting path algorithms for single-commodity flow problems in particular, the application of augmenting paths to multi-commodity flow problems has been sparse.

Rothfarb et al. applied augmenting paths in the following way [22]: To maximize multi-commodity flows with just one destination t, they added a logical super-source s, considered all commodities as the same commodity with new source s, and then solved the obtained single-commodity flow problem using the standard augmenting path method. Afterwards, the single-commodity flow is split into a multi-commodity flow again, using arc-chain decomposition. Since the arc-chain decomposition is independent of the initial flow, possibly all single-commodity flows are re-routed completely, even though already a small modification might have been sufficient. Furthermore, their algorithm does not deal with the problem of

Please cite this article in press as: S. Brandt, et al., Augmenting flows for the consistent migration of multi-commodity single-destination flows in SDNs, Pervasive and Mobile Computing (2016), http://dx.doi.org/10.1016/j.pmcj.2016.09.012