Computer Networks 77 (2015) 169–180

Contents lists available at ScienceDirect

Computer Networks

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

VHub: Single-stage virtual network mapping through hub location

Shashank Shanbhag [a], Arun Reddy Kandoor [b], Cong Wang [c], Ramgopal Mettu [d], Tilman Wolf [c,*]

[a] Oracle Corp., Santa Clara, CA, USA
[b] Google Inc., Mountain View, CA, USA
[c] ECE Dept., University of Massachusetts, Amherst, MA, USA
[d] CS Dept., Tulane University, New Orleans, LA, USA

ARTICLE INFO

Article history:
Received 25 June 2014
Received in revised form 6 November 2014
Accepted 21 December 2014
Available online 30 December 2014

Keywords:
Network virtualization
Network embedding
Hub location
Mixed integer programming
Traffic matrix

ABSTRACT

Network virtualization allows multiple networks with different protocol stacks to share the same physical infrastructure. A key problem for virtual network providers is the need to efficiently allocate their customers' virtual network requests to the underlying network infrastructure. This problem is known to be computationally intractable and heuristic solutions continue to be developed. Most existing heuristics use a two-stage approach in which virtual nodes are first placed on physical nodes and virtual links are subsequently mapped. In this paper, we present a novel approach to virtual network mapping that simultaneously maps virtual nodes and links onto the network infrastructure. Our VHub technique formulates the problem of mapping a virtual network request as a mixed integer program that is based on the *p*-hub median problem. Results from extensive simulations with synthetic and real virtual network requests show that our solution outperforms existing heuristics, including subgraph isomorphism backtracking search. Our approach requires fewer physical resources to accommodate virtual networks and is able to balance load more evenly across the network infrastructure.

1. Introduction

With the growing size and diversity of use of the Internet, it has become apparent that the use of a single, common network-layer protocol hampers the deployment of innovations in the network. The need to be compatible with the Internet Protocol (IP) has limited the deployment of new network protocols and new communication paradigms (e.g., content-based networking) to "islands" of deployments (e.g., IPv6) or end-system overlays (e.g., multicast).

Network virtualization is based on the premise that, rather than supporting a single (network-layer) protocol, the network infrastructure allows multiple protocol stacks to coexist in parallel. The idea of coexisting network slices was initially explored in the context of overlay networks (e.g., Planetlab [1]), but more recently has been proposed for the entire Internet. By virtualizing physical links and routers ("substrate"), multiple parallel networks ("slices") can be deployed *inside* the network and each slice can use its own set of protocols.

One of the appealing aspects of a virtualized network infrastructure is that it enables dynamic deployment (and removal) of network slices. This process matches well with economic models where customers request new

* Corresponding author at: 151 Holdsworth Way, Amherst, MA 01003, USA. Tel.: +1 413 545 0757.

E-mail addresses: shashank.shanbhag@gmail.com (S. Shanbhag), karunreddy30@gmail.com (A. Reddy Kandoor), cwang@ecs.umass.edu (C. Wang), rmettu@tulane.edu (R. Mettu), wolf@ecs.umass.edu (T. Wolf).

http://dx.doi.org/10.1016/j.comnet.2014.12.006
1389-1286/

virtual networks with specialized functionality from a virtual network service provider. These service providers have the ability to offer multiple, different virtual network services on their infrastructure (i.e., "network as a service" (NaaS)).

A key challenge is the management of resources in a virtual network infrastructure: When new virtual network slices are set up in a network, the virtual network provider needs to determine which physical resources are allocated to that slice. The resources that need to be considered include transmission bandwidth on links and processing resources on routers (for data plane and control plane protocol processing). This resource allocation problem is NP-hard since it can be reduced to the multi-way separator problem [2,3]. From the perspective of a virtual network provider, finding better solutions is critically important. The ability to determine resource allocations that meet customer needs and use less resources in the infrastructure directly translates into lower operational costs and/or the ability to accommodate more customers. Our work focuses on this mapping problem.

The main contribution of our work is VHub, a novel solution for determining mappings of virtual network requests to a given infrastructure. Our approach is to view the virtual network mapping problem as a *p-hub median problem*. Given a set of n nodes and a collection of source/destination flow requests, the p-hub median problem asks us to identify a subset of p hubs so as to minimize the cost of route each flow request through the chosen hubs. To map a given virtual network request on a physical network, we must select physical nodes and paths between them. Our approach is to model the virtual nodes as hubs and solve the associated hub location problem to compute the optimal placement of each request. Using this approach, we can formulate the constraints for nodes (e.g., geographic location, availability of processing resources) and for links (e.g., available bandwidth) in a single linear program whose solution directly translates to a mapping solution.

We extensively evaluate our approach and compare it to state-of-the-art virtual network mapping algorithms (DViNe [4] and vnmFlib [5]) using both synthetic virtual network requests and real request traces from Emulab. The results show that our VHub solution not only can find better solutions (e.g., accommodate up to 23% more virtual networks in a given physical network) but can do so 26–96% faster.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 provides a formal problem statement, Section 4 gives objectives and performance metrics, and Section 5 presents our VHub mapping method. Section 6 presents extensive evaluation results, where performance of different mapping methods are compared. Section 7 concludes the paper.

2. Related work

Network virtualization technology has been demonstrated and used in experimental platforms, including PlanetLab [1] and VINI [13], which provide a virtualization overlay on the Internet and testbeds (e.g., Emulab [14]). There have been efforts to federate existing testbeds into a larger experimental infrastructure that is based on virtualization (i.e., GENI [15] in the United States, OneLab [16] in Europe) as a precursor to the future Internet. The successful use of network virtualization in these testbeds supports the growing opinion that this technology may become fundamental to the Internet [17,18]. For a detailed survey on network virtualization, see [3,19].

In the control plane of a virtualized network, there is the need to support routing and other control functions in parallel (e.g., [20]). Beyond that, the management of virtual network slices within the infrastructure requires mechanisms for setup and tear-down of virtual networks at runtime, as well as monitoring of the operation of virtual network slices. In this paper, we address the key problem of virtual network mapping, in which we must allocate nodes and links of a virtual network to a physical substrate.

Algorithms for virtual network mapping presented in literature map virtual network requests to a physical network topology, but differ in the types of constraints that are considered. Constraints that are necessary in practice are: (1) limited bandwidth on links, (2) limited processing capacity on nodes (where protocol processing is implemented), and (3) geographic placement of nodes (so that users can access a virtual network). Mapping solutions that only consider bandwidth constraints were developed in other contexts (e.g., bandwidth resource allocation in virtual private networks [11]).

Table 1 presents a comparison of various mapping approaches. Emulab uses the *assign* algorithm [6] to map requests to the infrastructure. This algorithm uses simulated annealing to solve the virtual network mapping problem, but only considers bandwidth constraints on links. Constraints on node resources are limited to the condition of exclusive access to a particular physical node, or through resource sharing and isolation achieved through virtual machines. Fan and Ammar [7] also use simulated annealing for mapping traffic matrices to a physical network. The problem is restricted to allocating topologies to the physical network assuming no constraints on link bandwidth or node processing. Zhu and Ammar [8] use shortest path heuristics to solve the offline version of the mapping problem (i.e., where all requests are known in advance), also assuming infinite physical resources. Lu and Turner [9] try to solve the problem of mapping a single backbone-star virtual network topology onto a physical network with unlimited resources, also ignoring node processing constraints. Yu et al. [10] propose greedy heuristics for node mapping and multicommodity flow algorithms for splittable flows resulting in a two-stage mapping process that may possibly lead to sub-optimal solutions. R-ViNE [4] is a linear programming formulation for the two-stage virtual network mapping problem with bandwidth, processing, and node location constraints that uses multicommodity flow formulations for solving the edge mapping problem in the second stage. Lischka and Karl [5] propose vnmFlib, a variant of the best known backtracking algorithm to solve the subgraph isomorphism problem in a single stage [21] and also considering all relevant constraints. Cheng et al. [12] propose a Markov