Firewall Placement in Cloud Data Centers

Seungjoon Lee*, Manish Purohit†, and Barna Saha*

1 Introduction

As cloud data services proliferate, filtering the communication between different virtual machines in a data center becomes a necessity. Such filtering can be accomplished by placing firewalls at strategic nodes within the data center network and rerouting the communication flows to pass through a firewall. This abstraction introduces several basic location problems which arise in these contexts. Suppose a VM s wishes to send data to a VM t along path P. If there is no available firewall on path P, we need to reroute the data first from s to a firewall f and then from f to the destination t. Clearly, having too few firewalls would cause a large number of communication flows to be routed to a particular firewall leading to increased congestion in the links leading to the firewall. As latency in data centers is dominated by link congestion rather than distance, we focus on finding good firewall placements subject to a bandwidth constraint on links.

Related Work: In recent years, minimizing congestion in data centers has been an important research topic. A number of papers [2, 3, 4] consider the problem of "assigning" virtual machines to the physical servers, so that the congestion due to communication is minimized. Closer in spirit to our work is the Simultaneous Source Location (SSL) problem introduced by Andreev et al. [1] where each vertex has a demand D_v and the goal is to find the minimum number of sources so that each vertex receives a flow of D_v. SSL is a special case of our problem where each pair of communicating VMs resides on the same physical server and firewalls are uncapacitated.

* AT&T Labs - Research
† University of Maryland, College Park

SoCC'13, 1–3 Oct. 2013, Santa Clara, California, USA.
ACM 978-1-4503-2428-1. http://dx.doi.org/**10.1145/2523616.2525960**

2 Problem Definition and Results

Given a tree network $T = (V, E)$ with bidirectional edges having bandwidths $b(e)$, a set of communication flows (s_j, t_j, d_j) denoting a flow requirement of d_j units from leaf s_j to leaf t_j, and a capacity C denoting the maximum amount of flow a firewall can process, we wish to find the minimum number of firewalls necessary so that all flows can be feasibly routed through a firewall.

We consider two versions of the above problem, namely the soft-capacitated and hard-capacitated versions. In the soft-capacitated firewalls version, one is allowed to place multiple firewalls at a vertex. For this case, we show that an algorithm which considers vertices in a bottom-up fashion and makes local decisions at each vertex provides the optimal solution.

In the hard-capacitated version, however we are allowed to place at most one firewall at any vertex. This restriction makes the problem significantly harder and we develop two approximation algorithms for this version. Our first algorithm is based on reducing our firewall placement problem to the SSL problem described above. We extend the algorithm of [1] to handle sources with capacities and can prove the following theorem -

Theorem 1. *For the firewalls with hard capacities problem, if k is the optimum number of firewalls necessary, we obtain a solution with $\leq k$ firewalls such that the traffic through each edge is at most 3 times its bandwidth.*

Our second algorithm is a greedy algorithm that greedily picks the vertex that satisfies the greatest number of new demands. Using the submodularity of maximum flow, we prove that -

Theorem 2. *For the firewalls with hard capacities problem, the greedy algorithm is an $O(\ln(n))$ approximation.*

Ongoing work: In practice, software firewalls are often desirable due to their flexibility. However, software firewalls usually have an additional constraint that they can only be placed on the servers themselves (leaves). We are exploring the challenges posed by this restriction.

We have tested our algorithms on a production cloud data center and have obtained promising results.

References

[1] K. Andreev, C. Garrod, D. Golovin, B. Maggs, and A. Meyerson. Simultaneous source location. *ACM Transactions on Algorithms (TALG)*, 2009.

[2] N. Bansal, K.-W. Lee, V. Nagarajan, and M. Zafer. Minimum congestion mapping in a cloud. In *PODC 2011*.

[3] D. Dutta, M. Kapralov, I. Post, and R. Shinde. Embedding paths into trees: Vm placement to minimize congestion. In *Algorithms–ESA 2012*.

[4] X. Wen, K. Chen, Y. Chen, Y. Liu, Y. Xia, and C. Hu. Virtualknotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter. In *ICDCS-2012*.