A Cut-and-Solve based approach for the Virtual Network Function Assignment Problem

Presented by
Sara Ayoubi, Ph.D.
Concordia Institute for Information Systems Engineering
July 2016

Email: sa_ayou@encs.Concordia.ca
Outline

- Introduction
  - What is Cloud Computing?
  - The impact of virtualization on cloud data center network
  - The role of middleboxes in the cloud infrastructure

- Problem Motivation
  - Hardware-based Vs. Software-based (Virtualized) Middleboxes

- The Virtual Network Function Assignment Problem

- Proposed Work: A Cut-and-Solve Based Approach

- Numerical Analysis

- Conclusion
INTRODUCTION
What is Cloud Computing?

Benefits of Cloud Computing:
- Pay-Per-Use
- Elasticity
- No-Upfront Investment
- Lower Cap-Ex
- Lower Op-ex
- Reduced Risk
- Geo Diversity

Cloud Services:
- iCloud
- Google’s App Engine
- Amazon EC2 Cloud
- Microsoft Azure
- IBM’s Smart Cloud
- Sales Force
A Brief History…

• John McCarthy in 1960s predicted the concept of “Utility Computing”.

• In 1990s, the term “Cloud” was used to describe ATM networks.

• In 1999, SalesForce pioneered the concept of SaaS.

• In 2006, Google’s CEO used the term “Cloud” to refer to the new business model of providing services over the internet.

• In 2006, Amazon launched EC2 for HaaS.
What is Cloud Computing?

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” – The National Institute of Standards & Technology
Data Center Network

- Internet

Layer 3
- Core Routers
  - Aggregate Switches
    - Top-of-Rack Switches
      - Server Racks
Key Enablers of Cloud Computing

- Advancements in processing and storage technologies.
- High-bandwidth availability
- Mega data centers built at low-cost locations
- Network Virtualization
Network Virtualization

Client 1

Client 2

Client 3

Virtualization

Physical Network
Virtual Network Embedding

Network Node
Facility Node
Virtual Machine
Bandwidth link
Substrate connection

Physical Network

Embedded Flow (virtual link)

Client

Cloud Provider

WB
APP
DB
Policy-Aware Traffic Steering

- Network Node
- Facility Node
- Virtual Machine
- Middlebox
- Bandwidth link
- Substrate connection

Embedded Flow (virtual link)

Client

Cloud Provider

Substrate Network

Concordia University
Engineering and Computer Science
Middleboxes

- Ubiquitous elements in operator’s networks
- Enhance performance and security of tenant’s services; E.g.
  - Firewalls filter traffic based on predefined rules
  - Load balancers distribute traffic to multiple destination hosts
  - Intrusion Detection Systems collect data for security checks
- Service Function Chaining: Flows traversing multiple middleboxes in a predefined order.
Disadvantages of Middleboxes

- Run on specialized hardware
- Expensive and vendor specific
- Long time-to-market
- Placed at fixed location in the network
Network Function Virtualization

- Decouple software from hardware -> Softwarized-Middleboxes
- Run atop commodity hardware
- Instantiated on demand
- Can be easily modified and updated.
- Short time-to-market
Traffic Steering via Hardware Middleboxes

<table>
<thead>
<tr>
<th>Network Node</th>
<th>Facility node</th>
<th>Middlebox Type</th>
<th>Ingress Node</th>
<th>Egress Node</th>
<th>Physical link</th>
<th>Virtual connection</th>
</tr>
</thead>
</table>

Flow Request

Substrate Network
Traffic Steering via Virtualized Network Functions (VNF)

Flow Request

Substrate Network
THE VNF ASSIGNMENT PROBLEM
The VNF Assignment Problem

Major Concerns:

• Where to instantiate VNFs?
• How many?
• Which subset of VNFs each flow will be assigned to?

Why are these major concerns a concern?

• Limited number of VNF instances
• Limited network resources

“With Great Power Comes Great Responsibility”
Network Model Overview

1. Substrate Network $G = (N, L)$
2. VNF Types, denoted as $M$; where each $m \in M$ has:
   a. Limited number of instances $k_m$
   b. Resource Demands $w_m$
   c. Processing Capacity $p_m$
3. Traffic Flows denoted as $F$; where each $f \in F$ has:
   a. Ingress (origin) and egress (destination) node
   b. Policy chain $S_f$ of one or many VNF types
   c. Virtual Links $E_f$: pair of VNFs in $S_f$
   d. Bandwidth demand $b_f$
Problem Definition

**Problem Definition:** Given a substrate $G_s = (N,L)$ and a set of flows $F$, each with a forwarding policy $s_f$, find the optimal placement of VNFs that maximizes the number of admitted traffic flows, while respecting the capacity constraints of the substrate network.

- The VNF Assignment problem is NP-Hard [1]
- Logically divided into three sub-problems:
  1. VNF Mapping: $M \rightarrow N$.
  2. Flows-to-VNF assignment: $F \rightarrow M$
  3. Policy-Aware Traffic Routing: $F \rightarrow L$
Mathematical Formulation

Parameters:

- $G^s(N, L)$: Substrate network with $N$ nodes and $L$ links.
- $c_n$: the capacity of substrate node $n$.
- $b_{i,j}$: the capacity of substrate link $(i,j)$; where $i$ and $j$ denote the source and destination of the link, respectively.
- $F$: the set of flows, where every flow $f \in F$ is abstracted as a virtual graph $G^v_f(V_f, E_f)$, with a demand $\hat{b}_f$. Each virtual link $e \in E_f$ is composed of a pair of VNF types, where $o(e)$ and $d(e)$ denote the source and destination VNF types of edge $e$, respectively.
- $M$: the set of VNF types.
- $p_m$: the processing capacity of VNF type $m$.
- $w_m$: the resource demand of VNF type $m$.
- $K_m$: the maximum number of instances allowed for any VNF type $m$. 
Mathematical Formulation

Decision Variables:

- \( x_{m,n}^k = \begin{cases} 1, & \text{if instance } k \text{ of } m \text{ is placed on } n. \\ 0, & \text{otherwise} \end{cases} \)

- \( \delta_{m,n}^{f,k} = \begin{cases} 1, & \text{if } f \text{ is assigned instance } k \text{ of type } m \text{ on } n. \\ 0, & \text{otherwise} \end{cases} \)

- \( a_f = \begin{cases} 1, & \text{if flow } f \text{ is admitted.} \\ 0, & \text{otherwise.} \end{cases} \)

- \( y_{i,j}^{e,f} = \begin{cases} 1, & \text{if } e \in E_f \text{ is routed through link } (i,j). \\ 0, & \text{otherwise.} \end{cases} \)

- \( t_{i,j} \): indicates the amount of traffic measured on link \((i,j)\).
Mathematical Formulation

Integer Linear Program (ILP)

Max \( \sum_{f \in F} a_f \)

Subject to

VNF Placement

Flow-to-VNF Assignment

Policy-Aware Traffic Routing

\[
\sum_{j: (i, j) \in L} y_{i, j}^{e, f} - \sum_{j: (j, i) \in L} y_{j, i}^{e, f} = \sum_{k=1}^{K_m} \delta_{o(e), n}^{f,k} - \sum_{k=1}^{K_m} \delta_{d(e), n}^{f,k} \\
\forall e \in E_f, f \in F, i \in N.
\]

(7.9)

\[
t_{i, j} = \sum_{f \in F} \sum_{e \in E_f} y_{i, j}^{e, f} \hat{b}_f \quad \forall (i, j) \in L
\]

(7.10)

\[
t_{i, j} \leq b_{i, j} \quad \forall (i, j) \in L
\]

(7.11)
Related Work

- Existing work consist of ILP => Hard to scale
  - ILP requires 1 hour and 25 minutes to find the optimal policy-aware traffic steering for 35 flows.
  - Heuristic-based approach => No guarantee on solution quality
    - Place VNFs along the shortest path between ingress & egress node [2]
    - Solve the VNF placement and Traffic Steering disjointly [3]
    - Solve the VNF assignment sequentially for every flow using Viterbi [1]
A CUT-AND-SOLVE BASED APPROACH
Cut-and-Solve Technique

- Cut and Solve is a decomposition method:
  1. Master Model \(\Rightarrow\) Upper Bound
  2. Subproblem \(\Rightarrow\) Lower Bound
- Piercing cuts are extracted from the subproblem and introduced as new rows (constraints) in the master model
Solving the VNF Assignment Problem using Cut-and-Solve

\[ O = \]

Subject to

\[ \text{Max } \#\text{Flows} \]

\[ \text{Max } \#\text{VirtualLinks Routed} \]

Subject to

VNF Placement

Flow-to-VNF Assignment

Policy-Aware Traffic Routing

Master Model

SubProblem

Piercing Cuts
Cut-and-Solve Approach

Master Search Space

Cut 1
Cut 2
Cut 3
Piercing Cuts for the VNF Assignment Problem

Run Master Model

Add Diversification Cut

Run Sub-Problem

Add Separation Cut

O > P

yes

yes

no => Optimality
Separation Cut

- Change the physical host (placement) of at least one of the following:
  - The source VNF of e2
  - The destination VNF of e2
  - The source VNF of e4
Diversification Cut

f1

f2

Bottleneck Links
NUMERICAL ANALYSIS
## Performance Evaluation

<table>
<thead>
<tr>
<th># Flows</th>
<th>ILP Model</th>
<th>Cut-and-Solve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Admission</td>
<td>Runtime (s)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.67</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>1.57</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>5.32</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>9.99</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>41.91</td>
</tr>
<tr>
<td>30</td>
<td>28</td>
<td>46.33</td>
</tr>
<tr>
<td>35</td>
<td>32</td>
<td>4506.32</td>
</tr>
</tbody>
</table>
COMPARATIVE ANALYSIS

(a) DC \((N=36, L=48)\)

(b) \(R_1 \ (N=40, L=75)\)

(c) \(R_2 \ (N=60, L=160)\)
Conclusion

- We introduced the VNF Assignment problem.
- We presented how existing work consists of:
  - ILP-based formulations that do not scale.
  - Heuristics with no guarantee on the quality of the solution.
- We presented our Cut-and-Solve approach by showcasing:
  - A master model
  - A subproblem
  - Piercing Cuts: a Separation and a Diversification Cut.
- Through numerical analysis, we showed that our approach is 700 times faster than ILP-based formulations.
References


