Specifying and Placing Chains of Virtual Network Functions

Sevil Mehraghdam, Matthias Keller, and Holger Karl

2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)

Speaker: Tao Gao

2018-02-16

Group Meeting Presentation

Virtual Network Functions (VNFs)

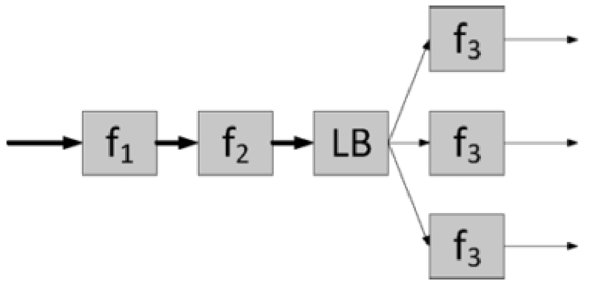
- Traditionally, network functions (e.g. load balancers, firewalls, and Intrusion Detection Systems (IDSs)) are implemented on **physical middle-boxes**. Middle-boxes are based on special purpose hardware platforms that are expensive and difficult to maintain and upgrade.

- Following the trend of virtualization in large-scale networks, network function that were deployed as middle-boxes are also being replaced by Virtual Network Functions (VNFs).

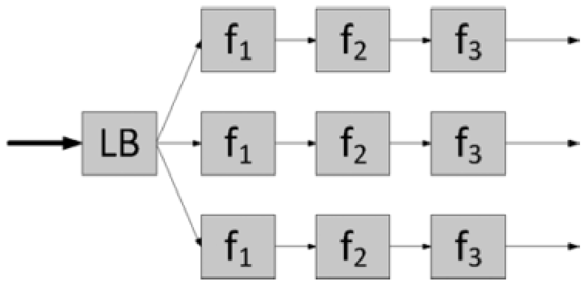
- Typically, network flows go through several NFs. This notion is known as *network function chaining* or *network service chaining*.

- NFs can modify the traversing network flows in different ways. There can also be a **dependency** among a set of NFs that should be applied to the traffic in a network, which requires special attention to the order of traversing the functions in chaining scenarios.

Chains of VNFs



(a)



(b)

- Depending on how each function in the chain modifies the data rate of the flows, different chaining options can have different impact on the traffic in network links, on application performance, or on latency. There are two challenges in this area.

  - ❑ Formalize a request for chaining several NFs together, while considering the possible dependencies among them.
  - ❑ Find the best placement for the functions, considering the requirements of individual requests and the overall requirements of all network applications and the combination of requests.

UCDAVIS NETWORKS RESEARCH LAB

Processing Deployment Requests

- Chaining requests are decomposed into different modules to build VNF graphs.
  - ✓ an individual function or a start/end point for the chain
  - ✓ optional order module
  - ✓ split module
  - ✓ parallel module
- Different modules have different permutation of the set of functions
  - ✓ For example, if a chaining request contains one optional order module with 3 different VNFs and one parallel module in which 4 VNFs can be placed with an arbitrary order, a total of $3!\cdot4!=144$ combinations of these modules is possible.

Processing Deployment Requests

Sort the functions in ascending order according to their ratio of outgoing to incoming data rate

↓

The function that reduces the data rate of the flows the most is placed before all other functions in the module.

↓

Get a VNF graph. Input to be deployed into the operator's network.

UCDAVIS NETWORKS RESEARCH LAB

Placement Of Chained Network Functions

- Input to the placement step is the capacity of network nodes and links, requirements of different network functions, and the combined VNF graph from the request processing step.

f	Instance of NF, which is mapped to a node of operator's network. A NF need some data center computational resources or/and switch computational resources; has an limited number of copies; can handle a certain number of requests
u	A deployment request that require a NF. A VNF graph consists of a set of *u*, which form different pairs (*u*,*u'*). A pair has a data rate requirement.
a	A start or an end point of a flow. Paths(*a*, *a'*) traverse different NFs. A path has a latency requirement.

Placement Of Chained Network Functions

Constraints

1) *Network Function Placement Constraints:*

   Placing functions in network nodes and mapping requests for using instances of network functions to these nodes;

2) *Path Creation Constraints:*

   Creating paths between functions;

3) *Metrics Calculation Constraints:*

   Collecting metric values, which are used for different objectives.

Placement Of Chained Network Functions

Objectives

1) Maximizing the remaining data rate on network links:

$$\text{maximize} \sum_{(v,v')\in E, v\neq v'} \text{remdr}_{v,v'}$$

2) Minimizing the number of used nodes in the network:

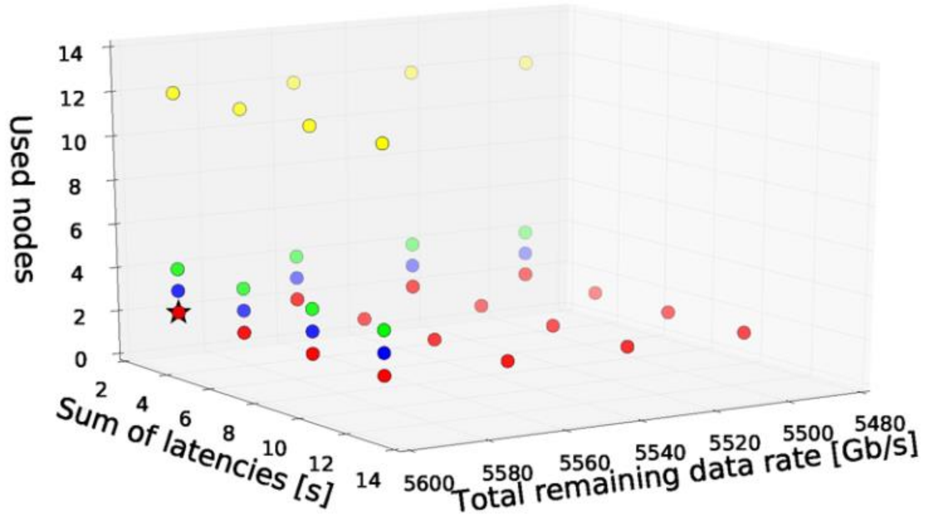
$$\text{minimize} \sum_{v\in V} \text{used}_v$$

3) Minimizing the latency of the created paths:

$$\text{minimize} \sum_{(a,a')\in l_{\text{req}}} \Big(\sum_{P\in \text{paths}(a,a')} \Big(\sum_{(u,u')\in P} \text{lat}_{u,u'} \Big)\Big)$$

UCDAVIS NETWORKS RESEARCH LAB

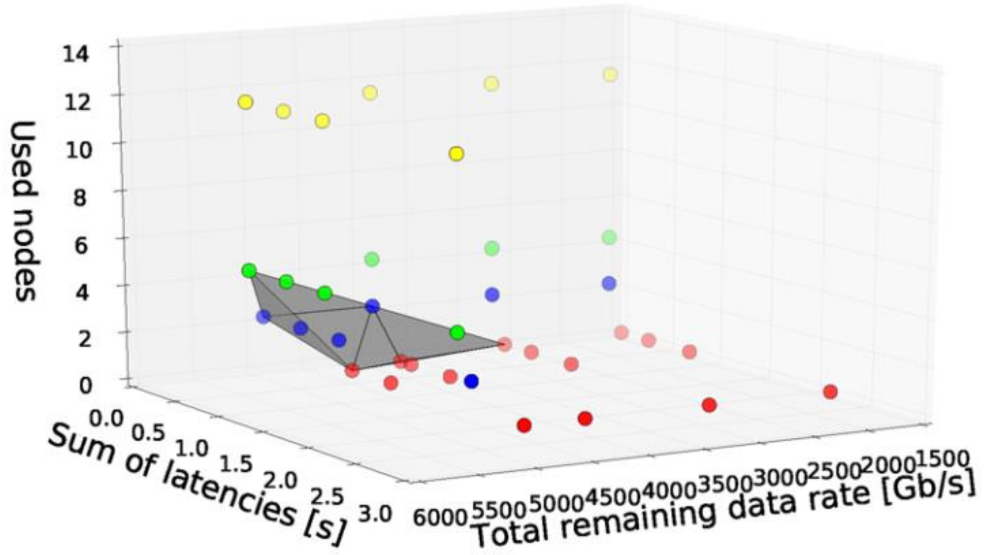
Results



(a) Results for a sample request set with optimal solution

- Results for different combinations of requests.
- There is a solution that is optimal in terms of all three metrics (the solution marked by a star).

Results



(b) Results for a sample request set including a Pareto set

- Results using Pareto set analysis.
- The results are obtained based on the trade-off among different objectives (i.e. multi-objective optimization)

Analysis

1) *Dynamic traffic scenario:*

   Actually, in reality, the set of requests cannot be obtained in advance, how to map requests into to the operator's network one by one;

2) *Computational resource model:*

   Computational resources include CPU, memory, storage, etc.;

3) *Optimal results:*

   Dividing the problem into two steps may get sub-optimal results. But it is hard to get optimal results since each sub-problem is NP-hard.

Thanks!