Cost-based placement of vDPI functions in NFV infrastructures

Mathieu Bouet, Jérémie Leguay, Vania Conan

Cost-based placement of vDPI functions in NFV infrastructures

Mathieu Bouet[1] and Jérémie Leguay[2] and Vania Conan[1]

[1] *Thales Communications & Security, France*
[2] *France Research Center, Huawei Technologies Co. Ltd*

Network Functions Virtualization (NFV) is transforming how networks are operated and network services delivered. The network is more flexible and adaptable. In particular, virtual Deep Packet Inspection (vDPI) engines can be dynamically deployed as software on commodity servers within NFV infrastructures for incremental monitoring. For a network operator, deploying a set of vDPIs over the network is a matter of finding the appropriate placement that meets the traffic management and operational cost constraints (license fees, network efficiency or power consumption). In this paper, we formulate the vDPI placement problem as a cost minimization problem. We cast the problem as a multi-commodity flow problem. We then propose a centrality-based greedy algorithm and assess its validity by comparing it with the ILP optimal solution on random networks.

Keywords: Network management, Network Functions Virtualization (NFV)

1 Introduction

Deep Packet Inspection (DPI) is a technique that allows fine-grained real-time monitoring of flows and users activity in networks and IT systems. It consists in filtering network packets to examine the data part (and possibly also the header) of a packet flow, identifying traffic types, searching for protocol non-compliance, viruses, spam, intrusions, or any defined criteria. Originally implemented as hardware middleboxes to be installed in network infrastructures, DPI are evolving towards the Network Function Virtualization (NFV) paradigm : they are being virtualized and delivered as software bundles that can be deployed and used on demand and on commodity hardware platforms. This approach is strongly supported by Internet Services Providers at ETSI [C$^+$12] to build the so-called NFV infrastructures, which basically consist in cloud platforms that can host network functions running. These infrastructures are intended to be deployed at PoPs (Points-of-Presence) or local aggregation points in carrier networks.

With virtual DPI (vDPI) network operators are facing a new set of challenges. The key question they have to address is the issue of where to instantiate the vDPI functions in their network. For the operator this decision is the result of a compromise between several possibly conflicting goals : in some case all flows must be monitored, and the obvious choice is to deploy one vDPI on every node. However, to reduce the cost (computing power, license fees, network utilization or power consumption), it may be more efficient to deploy less vDPI instances and pay an additional cost for rerouting traffic through these nodes. In other cases monitoring a sample of flows should be sufficient, but here again one should decide how many vDPI instances should be deployed and on which nodes.

Very recently, Luizelli et al. [L$^+$15] have proposed a binary search heuristic to jointly place VNFs (Virtual Network Functions) and map service chains onto them without any ordering constraints. The objective is to minimize the number of VNFs while meeting end-to-end delay guarantees. Our contribution differs from this work in mainly two points : i) our formulation is cost-based to finely represent global resource usages, ii) we consider that the sequence of functions is known in advance, i.e. each flow has to be monitored by one vDPI, which gives a competitive advantage to graph based heuristics such as ours. Addis et al. [ABBS15] proposed an ILP model that considers the traffic compression induced by multimedia gateways or firewalls for the cost efficient placement of VNF chains. They apply linearization techniques to increase the execution speed of an ILP solver. We propose a heuristic, for vDPI, to cope with the intractability of such formulations.

In this paper, we provide a general formulation of the vDPI placement problem as viewed by the network operator. We define a cost function that evaluates the quality of any given placement of vDPI and corresponding routing of the flows. We then present a centrality-based greedy algorithm and assess its validity by comparing it with the ILP optimal solution on random graphs.

2 Problem description

The problem we address in this paper can be formulated as follows : for a given NFV infrastructure and a given traffic demand, find a vDPI engine deployment that minimizes the overall cost. This cost optimization problem is the result of a joint minimization between i) the cost of vDPI engines and ii) the cost of the overall network footprint induced by flow redirections through the vDPI engines, while integrating iii) different operational constraints. These costs are financial costs associated with deployed vDPI engines (e.g. license price, CPU utilization, energy consumption...) and the cost of network resources (e.g. total cost of network ownership, capacity of the network to absorb new traffic). Operational constraints may include management limits such as the maximum number of engines to be deployed, the maximum used bandwidth per link (to be able to absorb peaks) and the maximum unmonitored flows. In the rest of this work, we assume that a third-party software editor provides the operator a vDPI software with a licence-based cost model : the cost is function of the quantity of vDPI engines deployed on the NFV PoP.

There is a tradeoff between the two main objectives which are, on the one hand, minimizing the number of vDPI engines and, on the other, minimizing the network load. Indeed, all the flows have to go through at least one vDPI engine to be analyzed. When the number of vDPI engines is small the network paths tend to be elongated. Therefore, minimizing the number of engines increases the additional used bandwidth. On the contrary, minimizing the used bandwidth increases the number of vDPI engines to be deployed. This problem can be casted as a multi-commodity flow problem [EIS75]. Its formulation has to be extended to have a vDPI probe monitoring each flow [BLCC15] with the objective function is :

$$Minimize \sum_{i \in V} dpi_i * \omega_{dpi} + \sum_{(i,j) \in E, f \in F} f_{size} * (x^f_{i,j} + y^f_{j,i}) * \omega_{bw} \quad (1)$$

where ω_{dpi} is the cost of a vDPI, ω_{bw} the cost for using a unit of bandwidth and f_{size} the size of the flow f which goes through the sets of links $x_{i,j}$ and $y_{i,j}$.

The constraints are those of multi-commodity flow problem applied with an extension to support pseudo-flows (a flow is decomposed into two pseudo-flows : one from the source to the vDPI and one from the vDPI to the destination).

3 A centrality-based greedy algorithm

3.1 Heuristic description

Our main objective is to minimize the number of sites upon which vDPI probes are activated. To identify key sites where vDPI should be placed, we base our approach on nodes' centrality. Several types of centrality have been proposed and studied in graph theory as indicators which identify the most important vertices within a graph. One of the most used centrality metric is the betweenness centrality, which corresponds to the number of shortest paths that pass through a node. A node with high betweenness centrality has a large influence on the transfer of items through the network, under the assumption that item transfers follow shortest paths. We propose a new centrality metric, which is derived from the betweenness centrality. It combines two graphs, the first one being in our case the network topology $G_n(V, E_n)$ and the second one the traffic matrix $G_t(V, E_t)$. Our centrality for node i, namely $centrality_i$, is equal to total size of the flows in G_t which have their shortest path going through i in G_n. A node with such a high centrality carries a large amount of traffic and is thus a good candidate to deploy a vDPI.

The objective function to balance the two costs remains the same as Eq. 1. It corresponds to the sum of the cost of used network resources and the cost of vDPI licenses activated. The heuristic we propose consists in a greedy algorithm that, at each step, considers placing a new vDPI on the node that has the highest centrality until the global cost stops decreasing. It is described in Algorithm 1.

Algorithm 1 Pseudocode for the greedy placement algorithm.

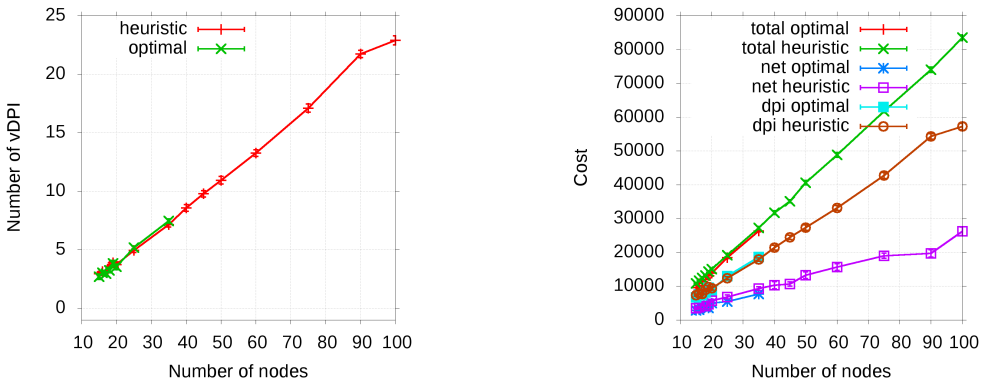
function GREEDY PLACEMENT (Topology graph : $G(V,E)$, List of traffic flows : F)
  $fit_{min} \leftarrow \infty$ ▷ best fitness value
  $F_u \leftarrow F$ ▷ list of unallocated flows
  $DPI \leftarrow \{\}$ ▷ list of selected nodes
  Compute nodes centralities in (G, F_u) (S1)
  $dpi \leftarrow$ the node with the highest centrality in (G, F_u)
  while $centrality_{dpi} > 0$ and $dpi \notin DPI$ **do**
    Add dpi to DPI
    Compute fitness value (S2)
    if $fitness(G, F_u, DPI) < fit_{min}$ **then**
      $fit_{min} \leftarrow fitness(G, F_u, DPI)$
    else
      break
    end if
    Remove from G the resources used by the flows in F_u whose shortest path goes through dpi **(S3)**
    Remove from F_u the flows whose the shortest path goes through dpi **(S4)**
    $dpi \leftarrow$ the node with the highest centrality in (G, F_u)
  end while
end function
return DPI

The network footprint is evaluated incrementally (initialization takes place before the first iteration). For each flow, its shortest path through the nearest vDPI is calculated taking into account the available resources. When this shortest path is found, the used resources are removed from the available resources and the next flow is considered. The total network footprint is equal to the sum of the used network resources. We adopt an iterative shortest path allocation of all the flows for computational efficiency reason. The computation of the network footprint relies on a stateful algorithm, keeping in memory the current flow allocation to speed up further computations. Indeed between one iteration and the next one, the list of selected vDPI engines only differs by the last element (since a vDPI engine is added at each iteration). Therefore, instead of recomputing all shortest paths through each vDPI probe for each flow, we just need to compute the shortest path through the last probe.

At all steps, if the fitness value, that is its total cost of a vDPI deployment, is higher than the one of the previous vDPI placement, the greedy algorithm terminates and returns the previous vDPI placement. Otherwise, the unallocated flows that traverse the new vDPI are considered as allocated. Their resources are removed from the available resources. The next iteration starts with an updated topology and a reduced quantity of unallocated flows. In addition to an increasing fitness value, the algorithm has two other termination conditions. It terminates when the highest centrality is equal to zero, which means that all the flows have been allocated to a vDPI, or when the new vDPI has already been chosen, which means that all nodes have a vDPI.

3.2 Complexity analysis

In Algorithm 1, steps (S1), (S2), (S3) et (S4) call the Dijkstra algorithm. More precisely, steps (S2) and (S3) call the evaluateNetworkFootprint() function, (see description in the previous section) whose complexity is $O(n^3 ln(n))$. Other calls (S1) and (S4) have a complexity of $O(n^2 ln(n))$ or less, and are experimentally negligible compared to evaluateNetworkFootprint(). If p is the probes count at the end of the execution, n the number of nodes in the graph and d the mean node degree, then, experimentally, with Erdos graphs (and $d = 3$) we have $p = n/3$. Therefore when all loops are combined we get a total complexity : $C(n) = 50 * n^4 * log(n)$ operations / comparisons for our heuristic.



(a) Number of vDPIs.

(b) Total cost.

FIGURE 1: Evaluation on random graphs with Erdos node degree distributions. Figures represent for a site opening costs of $2500 the number of vDPIs (left) and the costs (right).

4 Evaluation

We have implemented the ILP with GLPK and the heuristic using the Java Universal Network/Graph Framework (JUNG).We generated random graphs with the Erdös-Rényi degree distribution.The mean node degree is between 2.5 and 3. With this value, the resulting graph contained a big connected component and smaller ones, which are then connected to the big one by random links. We considered a traffic matrix where a 10 Mbit/s demand is created in both directions for each pair of nodes. We, of course, acknowledge that it does not represent reality, but aims at capturing an average case. We used a fix cost of $10 per Mb/s for the additional used bandwidth.

Each point on Fig. 1 corresponds to 10 simulations. We can observe on Fig. 1a that the heuristic find roughly the same number of vDPIs. The ILP stops after 35 nodes because of lack of memory while the same data the heuristic took 5.253s with our implementation. On Fig. 1b, comparing the optimal and the heuristic, we can observe a very good fit in terms of cost. In most of the cases, the heuristic finds an optimal solution.

5 Conclusion

In this paper, we formulated the vDPI placement problem as a cost minimization problem, capturing the different constraints the operator is facing. We casted the problem as a multi-commodity flow problem. We then proposed a centrality-based greedy algorithm and assessed its validity and comparing it to the linear program. In future work, we aim at reducing its complexity and address online placement.

Références

[ABBS15] B. Addis, D. Belabed, M. Bouet, and S. Secci. Virtual network functions placement and routing optimization. In *Proc. of 2015 IEEE Cloud Networking Conference (CLOUDNET)* , 2015.

[BLCC15] M. Bouet, J. Leguay, T. Combe, and V. Conan. Cost-based placement of vDPI functions in NFV infrastructures. *International Journal of Net. Management*, 25(6) :490–506, 2015.

[C+12] M. Chiosi et al. Network Functions Virtualization - An Introduction, Benefits, Enablers, Challenges and Call for Action, Oct. 2012.

[EIS75] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multi-commodity flow problems. In *FOCS*, pages 184–193. IEEE Computer Society, 1975.

[L+15] M. C. Luizelli et al. Piecing together the NFV provisioning puzzle : Efficient placement and chaining of virtual network functions. In *IFIP/IEEE Integrated Net. Management Symp. (IM)*, May 2015.