Server Placement and Assignment in Virtualized Radio Access Networks

Rashid Mijumbi*†, Joan Serrat*, Juan-Luis Gorricho*, Javier Rubio-Loyola‡ and Steven Davy†
†Telecommunications Software and Systems Group, Waterford Institute of Technology, Ireland
*Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
‡CINVESTAV, Tamaulipas, Mexico

***Abstract*—The virtualization of Radio Access Networks (RANs) has been proposed as one of the important use cases of Network Function Virtualization (NFV). In Virtualized Radio Access Networks (VRANs), some functions from a Base Station (BS), such as those which make up the Base Band Unit (BBU), may be implemented in a shared infrastructure located at either a data center or distributed in network nodes. For the latter option, one challenge is in deciding which subset of the available network nodes can be used to host the physical BBU servers (the placement problem), and then to which of the available physical BBUs each Remote Radio Head (RRH) should be assigned (the assignment problem). These two problems constitute what we refer to as the VRAN Placement and Assignment Problem (VRAN-PAP). In this paper, we start by formally defining the VRAN-PAP before formulating it as a Binary Integer Linear Program (BILP) whose objective is to minimize the server and front haul link setup costs as well as the latency (or distance) between each RRH and its assigned BBU. Since the BILP could become computationally intractable, we also propose a greedy approximation for larger instances of the VRAN-PAP.**

Keywords*—network function virtualization, server placement, resource assignment, virtualized radio access networks.

I. Introduction

The telecommunications sector continues to be faced with an apparent insatiable demand for higher data rates by subscribers. To keep up with this demand, new infrastructure is often needed, which leads to high CAPital EXpenses (CAPEX) and OPerating EXpenses (OPEX) for Telecommunication Service Providers (TSPs). This has significantly reduced profitability, and forced TSPs to find new ways of expanding the capacity of their networks while still remaining profitable [1]. Since the RAN takes up to 80% of CAPEX and 60% of OPEX, it has been identified as a candidate for achieving reduced CAPEX and OPEX. This can be achieved by decoupling some RAN functions from the physical devices on which they run so as to achieve better agility and efficient resource utilization. This is the concept of NFV [2] which is illustrated in Fig. 1.

The left part of Fig. 1 shows the current implementation of a RAN in which the RRH in each cell is associated with a dedicated BBU. In the VRAN scenario (the right part of Fig. 1), the servers responsible for the BBU functions are transferred to a shared physical infrastructure and virtualized. They are then connected to the RRHs over front haul links. The physical BBU servers may be located at a data center or distributed across the TSP's RAN node locations.

However, despite the promising gains, VRANs present some challenges especially with regard to front haul links, which are not only capital intensive, but also introduce high

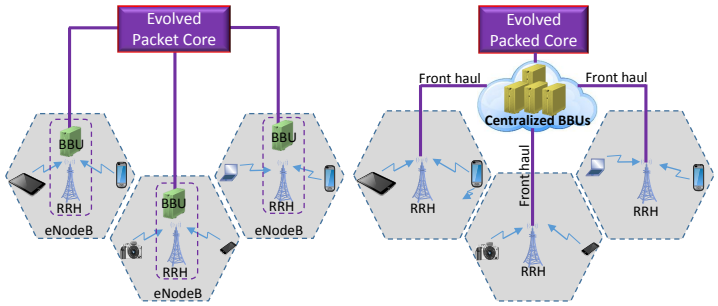

Fig. 1. Virtualization of LTE Radio Access Networks

latency [3]. In order to achieve the gains expected from VRANs, i.e. energy and spectral efficiency, front haul links must be able to provide a high bandwidth, at low capital and operating costs and low latency. In this paper, we propose algorithms for placing BBU servers and assigning RRH sites to them. In particular, we consider a TSP that already has an existing topology of BSs. To virtualize the RAN, the TSP should determine a subset of its existing network nodes where physical BBUs may be placed. Thereafter, for each of the RRHs at sites where no physical BBU has been placed, we should assign them to one of the placed BBUs. These two problems constitute what we refer to as the VRAN-PAP.

The rest of the paper is organized as follows: We discuss related work in section II before describing the VRAN-PAP in section III. In section IV, we formulate the VRAN-PAP as a BILP and propose a greedy approximation for it. The two algorithms are evaluated and discussed in Section V, and the paper concluded in Section VI.

II. Related Work

The placement of servers in VRANs is related to the Function Placement Problem (FPP) [4], [1]. The FPP involves the need to determine on to which physical resources (servers) network functions are placed. In the same way, the placement of servers is related to Virtual Data Center Embedding (VDCE) [5] and Virtual Network Embedding (VNE) [6], both of which are well studied problems. With regard to VDCE, most current approaches such as [7] focus on resource sharing through mapping Virtual Machines (VMs) to physical servers with the aim of improving server resource (e.g., CPU or memory) utilization, and maximizing the number of mapped VMs. Similarly, VNE deals with mapping nodes and links of a Virtual Network (VN) onto nodes and and links in a Physical Network (PN), with the objective of embedding as

978-3-901882-77-7

many VNs as possible onto a given substrate.

In addition, there have been efforts to enhance the efficiency of sharing the constrained and expensive front haul resources [3], [8]. However, most current approaches assume that the physical BBU servers have already been setup, and RRH sites assigned to them. The total number of physical BBU servers required to serve the whole RAN, together with their location relative to the RRH sites they serve is important since it does not only determine costs of server and front haul link deployment and maintenance, but also the resulting latency between each RRH and its assigned BBU server. Therefore, the design of efficient placement and assignment algorithms is an important initial step towards achieving the expected CAPEX and OPEX gains, as well ensuring that the resulting latency is acceptable. To the best of our knowledge, this is the first attempt to define and solve the VRAN-PAP in the context of NFV.

III. Problem Description

The VRAN-PAP considered in this paper is represented in Fig. 2. In the figure, a number of RRH sites are grouped together, as represented by cells with the same color. All RRH sites in a given group are *assigned* to a physical BBU server *placed* in one of the cells in that group. As an example, it can be seen from the figure that cells 1, 2, 3 and 4 share a common BBU server located in cell 3. For this group of RRHs, there are front haul links from cells 1, 2 and 4 to cell 3. In what follows, we propose a representation of the three main components of the scenario shown in Fig. 2.

A. RRH Sites

We consider a RAN with a set I of RRH sites that must be served by another set J of BBU servers, where $|J| \leq |I|$. The location of each RRH site $i \in I$ is given by $P_i(x, y)$, where x may be the latitude, and y the longitude. Each RRH site $i \in I$ provides service to U_i associated User Equipments (UEs) whose combined average processing requirements are δ_i. For each RRH site, we define a maximum desired level of latency τ_i, which gives a measure of the desired maximum distance from the RRH to the BBU.

B. BBU Servers

Due to budget restrictions, we consider that a TSP is able to set up a maximum of $p = |J| \leq |I|$ physical BBU servers to serve all the RRH sites in the RAN. Each physical server $j \in J$ has a maximum processing capacity of η_j. This maximum processing capacity is aimed at reflecting the fact that in practice, we would not be able to have a physical BBU server (or multiples of them in a single location) with unlimited processing capacity. Placing a physical BBU $j \in J$ involves a cost of f_j, which may be defined as a linear-cost function shown in (1). The cost is composed of two parts: a fixed initial cost u_j which takes care of the fixed investments such as space and installations, and a marginal/incremental cost v_j per unit of the processing capacity installed at the BBU server.

$$f_j = u_j + v_j \eta_j \tag{1}$$

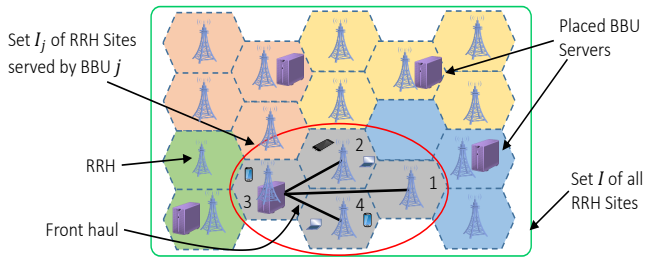

Fig. 2. Placement and Assignment Problem Representation

Each BBU server is located at the site of one of the RRHs. Each subset $I_j \subseteq I$ of RRH sites is served by a single BBU server j located in one of the RRH sites.

C. Front Haul Links

Each RRH site i is connected to the corresponding BBU server j by a front haul link $l_{ij} \in L$, where L is the set of all front haul links. Each front haul link $l_{ij} \in L$ has a latency t_{ij}, which is dependent on the distance d_{ij} between the site i and the server j and the speed of signals in the transport medium used. The cost of setting up a front haul link between a site i and a physical server j is c_{ij}, and is defined as a linear combination of an initial fixed cost ω_{ij} and a variable part dependent on the bandwidth B_{ij} required on the link as shown in (2). In turn, we define the bandwidth B_{ij} required for a front haul link between BBU j and RRH i as being directly related to the processing requirements δ_i of the RRH site, and is hence given by $B_{ij} = \gamma\delta_i$, where γ and χ are a constants.

$$c_{ij} = \omega_{ij} + \chi B_{ij} \tag{2}$$

For a given solution, the maximum latency that any RRH site would experience while communicating with its assigned physical BBU is a measure of the worst possible latency performance of the system.

IV. BILP Formulation

A. Definition of Decision Variables

Let y_j be a binary decision variable that takes on a value of 1 if a physical BBU server j should be setup, and 0 otherwise. In addition, we define x_{ij} as a binary variable that takes on value 1 if the RRH site $i \in I$ is assigned to a BBU server $j \in J$, and 0 otherwise. The task is to assign values to all possible occurrences of y_j and x_{ij}, such that both latency and total (server + front haul link) costs are minimized.

B. Objective

$$\text{minimize} \quad \alpha\left(\sum_{j\in J}\Big(f_j \times y_j\Big) + \sum_{i\in I}\sum_{j\in J}\Big(c_{ij} \times x_{ij}\Big)\right) + \beta\sum_{i\in I}\sum_{j\in J}\Big(t_{ij} - \tau_i\Big)x_{ij} \tag{3}$$

The objective in (3) is to minimize the latency as well as costs. In particular, the first term of the objective gives the

total costs of installing the BBU servers, while the second term includes the costs of setting up front haul links between each RRHs and the BBU that serves it. Finally, the last term gives the total deviation of the actual latency from the desired levels between each RRH and the corresponding RRH. The constants α and β may be used not only to give more importance either to costs or to latency, but also to scale the values of costs to be comparable to those of latency so as to have a meaningful summation.

C. Constraints

1) Placement and Assignment:

$$\sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I \tag{4}$$

$$x_{ij} - y_i \leq 0 \qquad \forall i \in I, j \in J \tag{5}$$

Constraint (4) ensures that each RRH is attached to one physical BBU server. This constraint ensures that each RRH is assigned to at least one eligible BBU by creating a front haul link between them. Constraint (5) ensures that a front haul link is created between a RRH site i and a BBU server j, only if j has been placed. Together, constraints (4) and (5) ensure that the required number of BBUs are placed to serve all RRH sites, and that RRHs are only assigned to sites where BBUs have been placed.

2) Resource Capacity and Budget Conservation:

$$\sum_{j \in J} y_j \leq p \tag{6}$$

$$\sum_{i \in I} \left(\delta_i \times x_{ij} \right) \leq \eta_j \qquad \forall j \in J \tag{7}$$

Constraint (6) ensures that the maximum number of BBU servers does not exceed the budget p, while (7) is a server capacity constraint that ensures that the total processing requirements of all RRHs assigned to a given BBU server do not exceed the actual physical resources installed at the server.

D. Greedy Approximation Algorithm

The BILP formulation in (3) - (7) could be solved by using one of the available commercial BILP solvers. However, the formulated BILP can be reduced to some known NP-Hard problems. For example, if we simplify the objective function and eliminate some constraints and variables such as (4) and (7), the resulting sub-problem can be reduced to the Maximal Covering Location Problem (MCLP) which is known to be NP-hard [9]. By extension, this implies that the much harder BILP in (3) - (7) would be computationally intractable for networks of realistic sizes.

In this subsection, we propose a Cost-Aware Greedy Algorithm (CAGA) that is more computationally viable for larger instances of the problem. CAGA is aimed at minimizing the combined cost of BBU servers and front haul links. CAGA works as follows: we start by sorting the possible BBU servers in ascending values of the installation cost. For each of the BBUs, we also sort the RRH sites according to the cost of creating a front haul link between the BBU and the RRH. Then, choosing the BBUs with the lowest cost, we assign RRHs to it starting with those with the lowest front haul link cost, until the processing capacity of the server is used up. This is repeated until either all the RRHs have been assigned, or the maximum BBU budget p is reached before assigning all RRHs, in which case the algorithm would have failed to find a feasible solution.

V. Evaluation

A. Simulation Setup

In order to evaluate the proposed algorithms, a simulator was programmed in Java. The RAN was generated on a 500 by 500 grid, with its topology determined using Brite with similar setting as those in [10]. Where no direct link exists between nodes i and j, the front haul link, and hence its cost c_{ij} are determined as a sum of costs of creating the front haul link along the shortest path over the RAN topology from i to j using Dijkstra's algorithm. We used the tool ILOG CPLEX 12.60 [11] to solve the BILP. Simulations were run on Windows 8.1 Pro running on a 16.00GB RAM, Intel i7 4.8GHz Processor Machine. The processing capacities of BBU servers are uniformly distributed between 250 and 500 units respectively, while the processing demand for RRH sites is uniformly distributed between 50 and 100 units. Each RRH site has a desired latency determined from a uniform distribution between 1×10^{-7} and $1 \times^{-6}$ Units. The fixed cost u_j or ω_{ij} of each BBU server or each hop of a front haul link respectively is 500 units. Constants η_j, γ and χ are set to default values of 1.

B. Results

Fig. 3 shows the variation of total BBU servers' and front haul links' cost with a changing size of RAN. As the number of RRH sites to be served increases, the costs increase. This is attributed to increases in the number of required front haul links as well as BBU servers to serve them. As expected, it can also be observed that the BILP performs better than CAGA. In fact, beyond a given number of RRH sites, it can be noted that the cost of CAGA reduces to zero. This is explained by the fact that at that point, CAGA is no longer able to find placement and assignment solution as the maximum allowed budget has been reached. For this simulation the maximum budget was 15 BBU servers. Fig. 4 confirms this by showing the actual number of utilized BBU servers. It is evident that CAGA reaches the maximum allowed number faster than BILP.

In Fig. 5, we represent resource utilization. This is defined

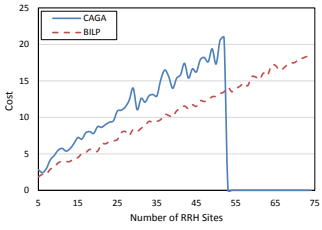

Fig. 3. Placement and Assignment Cost

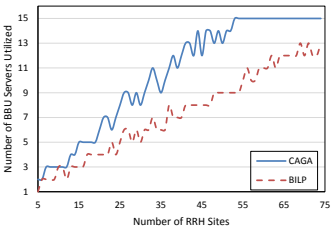

Fig. 4. Actual BBUs Placed

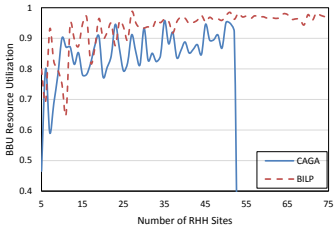

Fig. 5. BBU Average Resource Utilization

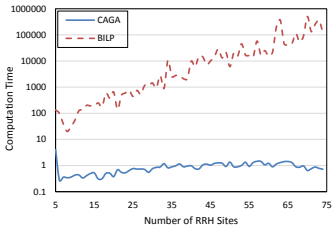

Fig. 6. Computation Time

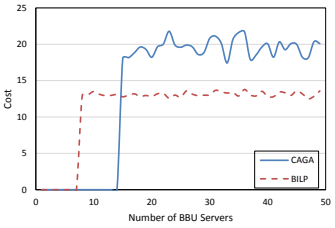

Fig. 7. Effect of Changing Budget

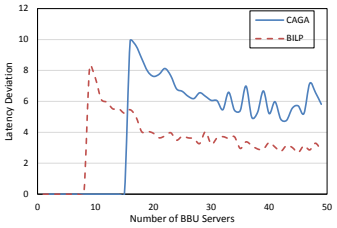

Fig. 8. RHH to BBU Latency

as the proportion of the total processing resources of all placed BBUs which have been assigned to all the RRHs assigned to them. We observe that while both approaches achieve a utilization ratio close to 1, that of BILP is not only slightly higher, but is also more stable. As expected, the resource utilization drops to zero when CAGA is not able to produce feasible solutions. Once again, the different profiles can be attributed to the superior efficiency of BILP which ensures that each placement and assignment utilizes the most appropriate of the available resources.

In Fig. 6, we represent (log scale) the computation times of the two approaches. Compared to BILP, it can be seen that the time required to find a solution using CAGA is not only lower, but also almost un affected by increasing the number of RRH sites. In fact, even when the algorithm is not able to find a feasible solution it utilizes a the same amount of time to determine this. This superiority in time complexity is not surprising since BILP is expected to become computationally intractable for larger problem instances. Therefore, Figs. 3 and 6 represent the trade-off that has to be taken either for solution quality or for faster solution. The decision on which of the two to utilize may depend both on network size as well as how often the placement and/or assignment has to be done.

Finally, Figs. 7 and 8 show the effect of varying the budget (maximum allowed number of BBU servers) on both the cost as well as latency. We define latency deviation as the average of all differences between the desired latency of a given RRH site and what is actually obtained after it is assigned to a given BBU. These simulations are performed for a fixed number of 25 RRH sites. We observe that initially, both cost and latency are zero. This is because a very low number of BBU servers does not produce enough resources to serve all the RRH sites. In addition, it can be observed that owing to its superior placement and assignment quality, BILP is able to produce feasible solutions earlier than CAGA. In addition, with regard to the latency, we observe that as the number of available BBUs increases, the latency reduces. This can be explained since when we have more BBU servers, there is more flexibility such that RRUs are assigned to those BBU servers closest to them.

VI. Conclusion

In this paper, we have formally defined the problem of placing and assigning resources in virtualized radio access networks. We also formulated a binary integer linear programming formulation of the same, and proposed a greedy algorithm for solving bigger instances of the problem. Simulations have shown that these two algorithms allow us to trade solution simplicity and enhanced computation time for better resource management. We hope that these algorithms could be used as a starting point to designing more advanced heuristics that share both the characteristics of solution quality and time complexity. In future, we hope to extend this work by use of more advanced heuristics to improve the solution quality while maintaining a low computation time.

Acknowledgment

This work is partly funded by the Science Foundation Ireland Research Centre CONNECT (13/RC/2077) and FLAMINGO, a Network of Excellence project (318488) supported by the European Commission under its Seventh Framework Programme.

References

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *IEEE Conference on Network Softwarization (NetSoft). University College London*, April 2015.

[2] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *Communications Surveys Tutorials, IEEE*, Accepted for Publication September 2015.

[3] C.-L. I, J. Huang, R. Duan, C. Cui, J. Jiang, and L. Li, "Recent progress on c-ran centralization and cloudification," *Access, IEEE*, vol. 2, pp. 1030–1039, 2014.

[4] H. Moens and F. D. Turck, "Vnf-p: A model for efficient placement of virtualized network functions," in *Network and Service Management (CNSM), 10th International Conference on*, Nov 2014, pp. 418–423.

[5] M. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 2, pp. 909–928, 2013.

[6] R. Mijumbi, J. Serrat, and J.-L. Gorricho, "Self-managed resources in network virtualisation environments," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, May 2015, pp. 1099–1106.

[7] M. Rabbani, R. Pereira Esteves, M. Podlesny, G. Simon, L. Zambenedetti Granville, and R. Boutaba, "On tackling virtual data center embedding problem," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, May 2013, pp. 177–184.

[8] J. Li, M. Peng, A. Cheng, Y. Yu, and C. Wang, "Resource allocation optimization for delay-sensitive traffic in fronthaul constrained cloud radio access networks," *Systems Journal, IEEE*, vol. PP, no. 99, pp. 1–12, 2014.

[9] R. Church and C. ReVelle, "The maximal covering location problem," *Papers of the Regional Science Association*, vol. 32, no. 1, pp. 101–118, 1974. [Online]. Available: http://dx.doi.org/10.1007/BF01942293

[10] R. Mijumbi, J. Serrat, J.-L. Gorricho, and R. Boutaba, "A path generation approach to embedding of virtual networks," *Network and Service Management, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[11] "IBM ILOG CPLEX Optimizer," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/about/, 2015, Accessed: 2015-07-19.