Mobility-aware VNF Placement in the LTE EPC

Akanksha Patel and Mythili Vutukuru
Indian Institute of Technology Bombay
{akankshapatel,mythili}@cse.iitb.ac.in

Dilip Krishnaswamy
IBM Research, Bangalore
dilip@ieee.org

***Abstract*—Traditional network infrastructure is facing significant challenges to support the growing bandwidth, coverage, and latency requirements of users, and to support new 4G and 5G use-cases. As a solution, telecom operators are looking to implement Network Function Virtualization (NFV) in future networks. The placement of Virtual Network Functions (VNFs) plays a critical role in reducing latency to serve users, while also helping in reducing the overall operational cost of the network. This work focuses on the VNF placement problem in the context of mobility and handovers, in scenarios such as mobile users travelling at high speeds, users served by small cells, and mobile IoT devices. We have modelled the VNF placement problem as an optimization problem that aims to reduce the latency in handovers resulting from mobility. We have compared our approach with other approaches, and we have demonstrated upto 60% reduction in the average time taken to serve handover requests in the network with no considerable increase in overall operational cost. We also provide a sub-graph based approach to solve for larger topologies and demonstrated its scalability.**

I. Introduction

Network functions virtualization (NFV) is transforming traditional networks by converting network functions into software appliances called virtual network functions (VNFs). These VNFs can be instantiated and removed dynamically at different nodes in the network based on the current traffic. In emerging 4G and 5G networks, distributed compute, storage and networking resources are expected to be available across nodes in core and edge networks; with probably less resources at the edge than at the core. Different VNF instances require different amount of resources based on user traffic they serve, and are to be instantiated at these nodes under such resource constraints using a VNF placement algorithm.

Our work addresses the problem of VNF placement for the LTE packet core. The LTE packet core has key network functions that need to be supported, such as MME (Mobility Management Entity), SGW (Serving Gateway), PGW (Packet Data Network Gateway), and eNodeB (Evolved NodeB). MME provides control plane support for access, mobility and authentication. SGW serves as the anchor for forwarding user plane packets whereas PGW providers support for connecting the user traffic to an Internet gateway. These functionalities can be provided by their respective VNFs. Note that, since eNodeB is involved in physical layer interactions with the UE (User Equipment), only a part of the processing on ENodeB can be virtualized as eNodeB VNF. VNFs can be replicated and instantiated across different nodes in the network, and different users can be mapped to different instances of these functions. Based on the placement of these functions in the network, users can experience different latencies for their interactions in the network.

In a typical LTE network, a large fraction (say, 70%) of users at any given time are static or pseudo-static, with the rest of the users being mobile or highly mobile. The number of mobile user devices such as smartphones, data cards, tablets, and the number of mobile IoT devices are increasing rapidly. Small cells are being increasingly used to provide better connectivity and bandwidth to users, and to increase the mean spectral efficiency in networks. As these devices move, the handoffs in the networks need to be managed quickly to enable a seamless experience for users. Thus, in emerging 4G and 5G networks, depending on the mobility of the user/devices and based on the limited compute, storage, and networking resource constraints across the nodes in the network, the selection of the location of the VNF instances to serve a particular user is an important problem to solve. In this work, we explore the placement of these VNFs across nodes in the network in a mobility-aware manner to reduce the latency involved in handovers in such scenarios. We formulated an Integer Program (IP) to solve this VNF placement problem, and compared it with other approaches. We demonstrate that our approach reduces the average time taken to process a handover request by upto 60%, while keeping the total operational costs low. To solve mobility-aware VNF placement problem on larger topologies where solvers take unacceptable time, we propose a topology partitioning method. We demonstrate that this technique can solve for topologies as large as 360 nodes in lesser time than the time taken by previous approach to solve for 28 nodes.

II. Related Work

VNF placement, also known as VNF embedding has been considered as an important problem in prior works, some of which are discussed below. [1] provides near optimal approximation algorithm for VNF placement in physical network with theoretically proven performance guarantees. The total cost here is computed as the sum of set-up costs of VNFs and the sum of distances between the clients and the nodes from which they get services. [2] solves the VNF embedding problem when the virtual network topology on which embedding needs to be done is not fixed. They provide approach to optimize virtual network topology and VNF embedding simultaneously for minimizing the cost of occupied links and node resources. [3] proposes approach for VNF placement on heterogeneous servers to maximize the total throughput of the system. [4] mainly discusses PGW instantiation and placement problem in Carrier Cloud to minimize the overall cost to the network operator, while ensuring QoE. [5] proposes Integer Linear Programming model and heuristics that minimize the number of VNF instances mapped on the infrastructure. [6] discussed policy-aware placement of VNFs in hierarchy of datacenters for minimizing cost of operating the datacenters while considering the relative priority of VNF placements at a data center and the desired performance requirements. [7] is a survey describing other such approaches. Although many of these approaches aim to minimize the overall cost on operator, considering different scenarios, objectives and constraints, none of them considers handover latencies. In this work we aim to solve the EPC VNF embedding problem for reducing the total time needed for handovers, especially meant for networks that experience a large number of handovers, e.g., users moving between large number of small cells or users in high speed transport.

There are other orthogonal directions of work that aim to make handovers efficient. [8] is a review of prior works that optimize the process of finding potential networks for handover, handover decision making algorithms for selecting network for handover, and strategies to execute handover in heterogeneous wireless environment. [9] is another direction of research that proposes a novel distributed mobility management approach for optimizing handover efficiency in flat architectures. [10] proposes use of caches for reducing redundant information exchange during handovers in LTE when performed on PON back-hauls. Such approaches can be used along with the proposed approach to make handovers more efficient.

III. Problem Description

We begin by describing the inputs to our model, summarized in Table I, then discuss how we model user requests including handover requests. Finally, we describe the necessity to perform mobility-aware instantiation of VNFs in the system.

A. Topology

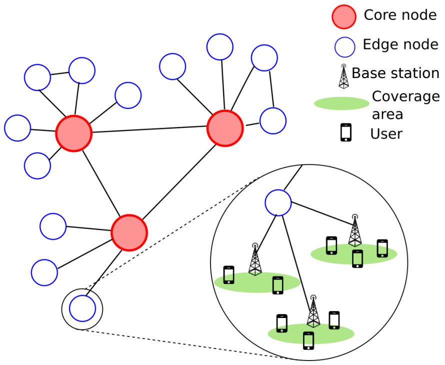

Fig. 1. Physical substrate network of a telecom operator

We envision a substrate network of interconnected processing nodes, each of which is either a server, a rack of servers or a data center. Each processing node can either be a core node or an edge node, as shown in Figure 1. Usually, the network has a large number of edge nodes, which are located in proximity of users and have limited capacities. In contrast, there are lesser core nodes and they have larger capacities. Users are connected to base stations (like pico-cell base stations or eNodeBs), which are connected to edge nodes. Multiple edge nodes are connected to a core node and some edge nodes within a core node may be connected among themselves. Some of the core nodes connect to a Packet Data Network (PDN), and act as gateway that connects users to external networks. Each processing node has limited processing resources, and a cost involved for processing requests. For simplicity, we considered only processing resources and assumed we are not limited by link capacity in our analysis. However, we do assume that there is some cost associated with using them so that we can use a lower / higher cost for links with higher / lower bandwidth respectively. We assume that for communication between any two nodes, the shortest delay path, computed from the knowledge of link delays, is always used. The sum of delays in the shortest delay path connecting nodes n_a and n_b is given as $\delta Link_{n_a,n_b}$, and the sum of the cost of using physical links in this path as $\kappa Link_{n_a,n_b}$.

B. User requests

Users can send different kinds of requests to the VNFs like requests to connect to the network, to detach from the network, to handover to another base station, or to forward user plane packets through the EPC. The node at which the requests of a user arrive is called the ingress node. Each ingress node, which is an edge node, receives requests from all users served by base stations attached to itself. Each kind of user request at an ingress node has a designated egress node, where the response packet or user's packet leaves the network. For instance, in LTE, the attach request is responded through the same ingress edge node, because the attach response is sent back to the user. However, a user plane request for forwarding packets to an external network is sent to a gateway core node, hence has a different egress node.

Requests of different types are served by different sequences of Virtual Network Functions (VNFs), known as service chains. We model any request as a combination of the amount of processing (CPU cycles) required at each network function in the service chain and the amount of data transferred (sum of packet sizes) between network functions to serve the request. Also, the bytes transferred between VNFs, say v_1 and v_2 to serve request r is the sum of size of packets transferred between the two VNFs, irrespective of the direction of packet transfer (denoted by $\beta RVV_{r,v1,v2}$). For instance, consider the service chain of attach request in LTE, shown in Figure 2. The attach request can be modelled as (i) processing required at the ingress node v_i, MME, SGW, PGW, and the egress node v_e, and (ii) communication between VNFs (v_i, MME), (MME, SGW), (SGW, PGW), and (MME, v_e). Serving a request incurs two kinds of costs, (i) processing cost, which is the sum of costs of processing at nodes where VNFs are instantiated, and (ii) communication cost, which is the sum of costs of using links for sending packets among VNFs that

TABLE I
INPUT SETS, PARAMETERS, AND DECISION VARIABLE

Set	Meaning
PN	set of processing nodes
VN	set of VNFs
RT	set of request types
TC	set of traffic classes
Parameter	**Meaning**
n	index for processing nodes
t	index for traffic classes based on tuples (ingress node n, mobility μ)
v	index for VNF types
r	index for user request types
m_t	$\in [1, \infty)$, mobility class of users in traffic class t
$IsIn_{t,n}$	1, if all requests of traffic class t enter network at node n; 0, otherwise
$IsGW_{t,n}$	1, if n is the gateway node to PDN for traffic class t; 0, otherwise
$IsIE_v$	1, if VNF v denotes an ingress/egress VNF; 0, otherwise
$AtGW_v$	1, if VNF v shall be placed at gateways; 0, otherwise
ρN_n	processing power (GHz) at processing node n
$\rho TV_{t,v}$	processing (cycles) reqd at VNF v to serve requests of traffic class t
$pktRVV_{r,v1,v2}$	packets transfer b/w VNFs $v1$ and $v2$ to serve req. r
$\beta RVV_{r,v1,v2}$	bytes transfer b/w VNFs $v1$ and $v2$ to serve req. r
δR_r	time budget (sec) to serve request r
$\delta Link_{n1,n2}$	delay (sec) between processing nodes $n1$ and $n2$
κN_n	cost ($/GHz) for using processing node n
$\kappa Link_{n1,n2}$	cost ($/Mbps) b/w nodes $n1$ and $n2$
$\phi_{t,r}$	freq. of req. r (/sec) of traffic type t
$\phi ho_{t1,t2}$	frequency of handovers (/sec) from ingress node of traffic type $t1$ to ingress node of traffic type $t2$
$\beta ho_{v1,v2}$	bytes transferred between VNF $v1$ of source traffic class to VNF $v2$ of target traffic class
$pktho_{v1,v2}$	packets transferred between VNF $v1$ of source traffic class to VNF $v2$ of target traffic class
$TotalHo$	total handovers per unit time in the network
Decision Variable	**Meaning**
$loc^n_{t,v}$	1, if VNF v to serve traffic of class t is instantiated at processing node n; 0, otherwise

are instantiated at different nodes. We assume that the cost of transferring packets among VNFs within a node is negligible.

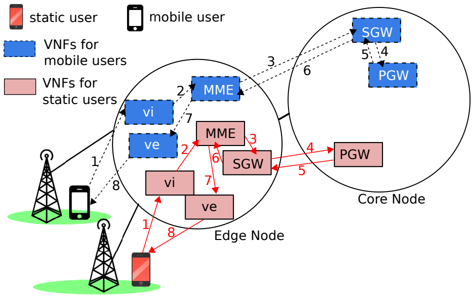

Fig. 2. Attach requests of users of different mobility at same ingress served by different VNF instances, based on their mobility classes (static and mobile)

C. Mobility and handovers

The mobility of a user can be defined in various ways according to the available parameters, such as the number of requests a user makes before making a handover request, the duration after which user requests a handover, or the speed of the user. Each user can belong to one *mobility class*, each class representing a range of mobility values. We divide users in terms of traffic class, which is a two-tuple <ingress node, mobility class>. This means that all requests arriving at a particular edge node, of users that have similar mobility, are assigned to one traffic class.

There can be several kinds of handovers at various levels of the system architecture. For instance, in LTE, there can be inter-eNB handovers, inter-MME handovers, inter-SGW handovers or even inter-RAT handovers. Handovers among base stations connected to the same edge node are simple to handle, because traffic from both base stations are handled by common VNFs as they belong to same traffic class. Similarly, inter-eNodeB handover in LTE is handled by the common MME and SGW. Such a handover can simply be handled like any other request, by considering amount of processing required at VNFs and the communication between the VNFs.

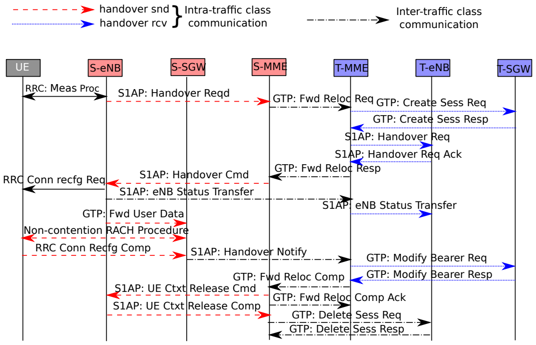

Fig. 3. Modeling of handover request in LTE

A different kind of handover occurs when the handover is between base stations connected to different edge nodes. In this case, the user moves from a base station served by VNF instances of one traffic class to a base station served by VNF instances of another traffic class. This is similar to an inter-SGW handover in LTE as shown in Figure 3. Any such handover request involves three communication domains: communication amongst VNF instances of the source traffic class, communication amongst VNF instances of the target traffic class, and communication between VNF instances of the source and the target (in Figure 3, between source MME and target MME). It involves two processing domains: processing at VNFs of source traffic class and processing at VNFs of target traffic class. Using the logic that a request can be modelled as processing and communication/packet transfer, we model the inter-traffic class handover requests as an aggregate of (i) `handover-snd` request at source (processing required at source VNFs and communication among them), (ii) `handover-rcv` request at target (processing required at target VNFs and communication among them), and (iii) communication between source and target VNFs. Due to this, the total number of requests at an ingress node are the sum of the number of requests except inter-traffic class handover requests, the number of `handover-snd` request for handover from the current node, and the number of `handover-rcv` requests for handover to the current node.

D. Mobility-Aware VNF Placement

If the VNF instances serving the source and the target base stations are same, then handovers can be quick. However, when VNF instances serving the source and the target base stations are different, then there are additional delays due to communication between the source VNFs and the target VNFs, which may be at different locations. One approach to reduce these delays is to place VNF instances serving all users in core nodes. However the same VNF instances also serve requests other than handover. Some of these requests may have a tighter time budget, that may not be fulfilled if all VNFs are at the core. Another fact is that static users do not need a handover. Therefore, placing all VNF instances serving static users at core nodes will only add to communication costs and unnecessary delays. Similarly, less mobile users need handovers less frequently as compared to highly mobile users. Thus, different VNF instances shall be instantiated for serving users with different mobility classes, attached to same ingress node, so that each of them can be placed at optimal locations according to the mobility class requirements.

We propose one VNF instance of each kind of VNF per mobility class per ingress node. We define this entity as a *traffic class*, a two-tuple <ingress node, mobility class>.

IV. Optimization Objective and Constraints

Given the information about the topology of the network, the incoming traffic mix of each mobility class arriving at each node, NFV specific information such as the amount of processing required by each request at each VNF, the amount of inter-VNF data to be transferred to serve a request, and the cost of using network resources, we formulate a model that outputs the optimal placement of VNF instances for each traffic class, that minimizes the average handover latency in minimum possible overall operational expenditure, while meeting the SLAs. We define the decision variable, $loc^{n}_{t,v}$, as the node n at which a VNF instances v that serves traffic class t shall be instantiated. We assume that one VNF instance per type is sufficient to serve users of a traffic class. We ensure through our model that the VNF instance in instantiated at a location where the required amount of processing resources are available. The amount of processing power to be assigned to each VNF instance of a traffic class is calculated based on frequency of incoming traffic of that class, and the amount of processing required by each request at each VNF instance.

A. System Optimization: Basic Integer Program

The **objective** of the model is to

$$minimize \quad AvgHoLatency$$

The average latency per handover ($AvgHoLatency$) is the sum of the latencies incurred in each handover divided by the number of handovers per unit time ($TotalHoLatency/TotalHo$). The $TotalHoLatency$ is the sum of latencies in transferring packets within the same traffic class for each handover and the sum of latencies in transferring packets between the source traffic class and the destination traffic class of each user.

$$\begin{aligned} TotalHoLatency = & \sum_{n_a,n_b \in PN} \sum_{t \in TC} \sum_{v_x,v_y \in VN} \sum_{r \in [ho-snd,ho-rcv]} \\ & (\phi_{t,r} * loc^{n_a}_{t,v_x} * loc^{n_b}_{t,v_y} * pktRVV_{r,v_x,v_y} * \delta Link_{n_a,n_b}) \\ & + \sum_{n_a,n_b \in PN} \sum_{t_1,t_2 \in TC} \sum_{v_x,v_y \in VN} (\phi ho_{t_1,t_2} * loc^{n_a}_{t,v_x} * loc^{n_b}_{t,v_y} \\ & * pktho_{v_x,v_y} * \delta Link_{n_a,n_b}) \end{aligned} \tag{O1-1}$$

Constraints: The model has to be solved to obtain optimal VNF embedding while considering following constraints.

Processing Constraint: The sum of the processing power allocated to VNF instances in a node should not be more than the available processing power at that node.

$$\sum_{t \in TC} \sum_{v \in VN} loc^{n}_{t,v} * \rho TV_{t,v} \leq \rho N_n \quad \forall n \in PN \tag{C1}$$

Delay Constraint: The delays in serving each request shall be less than the delay budget to respond to that type of request.

$$\begin{aligned} \sum_{v_x,v_y \in VN} \sum_{n_a,n_b \in PN} & (loc^{n_a}_{t,v_x} * loc^{n_b}_{t,v_y} * pktRVV_{r,v_x,v_y} * \\ & \delta Link_{n_a,n_b}) \leq \delta R_r \quad \forall r \in RT,\ t \in TC \end{aligned} \tag{C2}$$

Location Constraints: We categorize some VNFs as *ingress/egress* VNFs (IE) and some as *gateway VNFs* (GW). An *ingress VNF* receives all kinds of requests from users and delivers the requests to intended VNFs, such as to *MME* if it is an attach request and to *SGW* it is user instantiated data transfer request. Egress VNFs can be of different types such as an *MME-egress* that receives packets from *MME* and delivers them to the user. IE VNFs are low-resource consuming VNFs that provide the flexibility to place other primary resource-consuming VNFs at any node, as packets can be sent to their destination (user or external network) by these VNFs. A *gateway VNF* is an IE VNF that shall be located at the processing nodes which have connectivity to external network. In our model, for LTE, the *SGW-egress* is a type of *gateway VNF*. So, the first location constraint is that each gateway VNF for a traffic type shall be placed at the processing node assigned as gateway for that class.

$$\begin{aligned} IsIE_v * AtGW_v * IsGW_{t,n} * (1 - loc^{n}_{t,v}) = 0 \\ \forall t \in TC,\ v \in VN,\ n \in PN \end{aligned} \tag{C3}$$

Secondly, all IE VNFs, which are non-gateway VNFs, like *MME-egress* and *SGW-egress* shall be placed at the ingress processing node as they interact with users.

$$\begin{aligned} IsIE_v * (1 - AtGW_v) * IsIn_{t,n} * (1 - loc^{n}_{t,v}) = 0 \\ \forall t \in TC,\ v \in VN,\ n \in PN \end{aligned} \tag{C4}$$

Feasibility Constraint: One instance of each processing type is instantiated for each traffic class somewhere in the network,

$$\sum_{n \in PN} loc^{n}_{t,v} = 1 \quad \forall t \in TC,\ v \in VN \tag{C5}$$

B. Improved Model

The above model will produce a placement scheme which minimizes the average latency per handover. However, there can be several such placements schemes with same minimum average latency per handover. We modify our model to choose a scheme which costs the least amongst the possible minimum handover latency deployments:

$$minimize \quad M * AvgHoLatency + OverallCost \qquad \text{(O2)}$$

where M is a very large number and the $overallCost$ is calculated as the sum of: *processing cost*, *intra-traffic class communication cost*, and *inter-traffic class communication cost*. The *processing cost* is the sum of the product of the processing power allocated to a VNF instance at a location and the cost of using processing resource at that location:

$$\sum_{n\in PN}\sum_{t\in TC}\sum_{v\in VN} loc^{n}_{t,v} * \rho TV_{t,v} * \kappa N_n \qquad \text{(O2-1)}$$

The *intra-traffic class communication cost* is the sum of cost of transferring packets between every pair of VNF instances of the same traffic class:

$$\sum_{n_a,n_b\in PN}\sum_{v_x,v_y\in VN}\sum_{t\in TC}\sum_{r\in RT} \phi_{t,r} * loc^{n_a}_{t,v_x} * loc^{n_b}_{t,v_y} * \beta RVV_{r,v_x,v_y} * \kappa Link_{n_a,n_b} \qquad \text{(O2-2)}$$

The *inter-traffic class communication cost* is the sum of the cost of transferring packets between every pair of VNF instances of the different traffic class, primarily due to handover:

$$\sum_{n_a,n_b\in PN}\sum_{v_x,v_y\in VN}\sum_{t_1,t_2\in TC} \phi ho_{t_1,t_2} * loc^{n_a}_{t_1,v_x} \; loc^{n_b}_{t_2,v_y} * \beta ho_{v_x,v_y} * \kappa Link_{n_a,n_b} \qquad \text{(O2-3)}$$

For instance, in Figure 3, the *inter-traffic class communication cost* is the cost of communication between S-MME (the MME instance to serve source traffic class) and T-MME (the MME instance to serve target traffic class).

C. Working with Large Topologies

The proposed models can be run on the existing solvers like Bonmin solver [11] to compute the placement of VNFs in small topologies. As the number of nodes increase, the number of variables and constraints to be handled by the solver increase non-linearly. Therefore, solvers are unable to solve larger topologies in reasonable time. One of the motivations to use NFV is to be able to dynamically instantiate and remove VNFs with changes in user traffic. In such scenarios, less time-consuming approach is preferable. To solve for such a scenario, we propose dividing the topology graph into subgraphs. Each subgraph comprises a gateway node, the edges whose users use this gateway, the core nodes to which these edges are connected, and the connecting links. We assume that in a network, each user attached to ingress node has a predefined gateway from where the user traffic enters or leaves to the external network. The graph shall be divided into a set of subgraphs, one for each gateway. For instance, in Figure 4, graph is divided into subgraph 1 for gateway node 1 and subgraph 2 for gateway node 3. The improved model is then run for each subgraph, and the outputs will determine the placement of VNFs at nodes in the subgraph, and hence in the original topology graph as well.

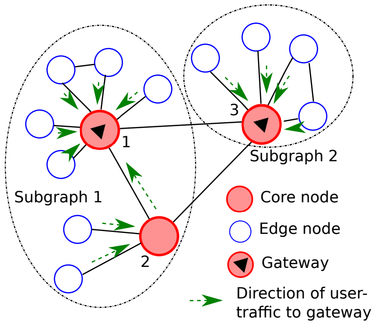

Fig. 4. Division of graph into subgraphs

V. Evaluation

We evaluated the performance of our models, (Ho, the model that minimizes average latency per handover, Opt, the improved model that selects the minimum average handover latency placement scheme which reduces the overall cost), and the subgraph approach ($SubGr$ that divides graphs into subgraphs and solves each subgraph separately) in the LTE scenario. We limited ourselves to three traditional VNFs—MME, SGW, and PGW, and four special VNFs—ingress, MME-egress, SGW-egress and PGW-egress. We considered following kinds of requests:

- *Initial attach request*: when a user whose information is not stored in the network tries to attach itself to the network. The processing of this request involves processing at MME, SGW, and PGW.
- *Data transfer request*: to forward user plane packets from user to external network, which involves SGW and PGW.
- *Handover request*: We considered inter-SGW handover request that involves processing at MME, SGW and PGW of both source and target traffic classes, as shown in Figure 3.

We used the Bonmin solver [11] for solving the integer program, and compared our models with two other approaches of VNF-placement.

- *Greedy Approach (Greedy)* places VNFs as close to ingress node as possible. This is a straw man approach that aims to minimize the communication cost by placing all VNF instances close to corresponding users.
- *Minimum Cost Approach (MinCost)* is an advanced approach that minimizes the overall cost (processing cost + communication cost). It however does not consider reducing the average latency per handover.

It should be noted that the $MinCost$ approach and the $Greedy$ approach do not have any notion of traffic classes, hence each VNF instance has been designed to serve requests of users of one ingress node. However, each VNF instance of Ho and Opt has been designed to serve users of one mobility type of an ingress node. Therefore, VNFs of $MinCost$ and

$Greedy$ require larger processing power at the location where they are instantiated, as compared to VNFs in our approaches, which serve only a part of user requests arriving at the ingress node based on the mobility type.

We evaluated the models for the following three metrics. Each data point in the output is generated by taking the median, the minimum and the maximum values of metrics in 20 random topologies. Each topologies has 20 nodes (including 5 cores) generated based on the values of parameters as given in Table II. Please note that, in graphs for following metrics, the plots of Opt, Ho, and $SubGr$ are overlapping, as they produce same outputs.

- *Number of packet transfers due to handover*: This is calculated as the sum of inter-traffic class packet transfer and intra-traffic class packet transfer due to handovers between different processing nodes. $TotalHoPkts =$

$$\sum_{n_a,n_b \in PN, n_a \neq n_b} \sum_{t \in TC} \sum_{v_x,v_y \in VN} \sum_{r \in [ho-snd, ho-rcv]} (\phi_{t,r} * loc^{n_a}_{t,vx} * loc^{n_b}_{t,vy} * pktRVV_{r,v_x,v_y}) + \sum_{n_a,n_b \in PN, n_a \neq n_b} \sum_{t_1,t_2 \in TC} \sum_{v_x,v_y \in VN} (\phi ho_{t_1,t_2} * loc^{n_a}_{t_1,v_x} * loc^{n_b}_{t_2,v_y} * pktho_{v_x,v_y})$$

  Figure 5 shows that the total number of packets transfers between processing nodes is reduced by upto 70% in *Opt*, Ho and $SubGr$ approaches as compared to the other approaches. $SubGr$ approach, though working on subgraphs, produces an optimal placement strategy to minimize the number of control packets in the network due to handovers. $Greedy$ and $MinCost$ approaches need higher processing power per location, hence cannot optimally place all VNFs to reduce the number of packets exchanged due to handovers, until edge processing power is high.
- *Average Handover latency*: This is calculated as in the objective function. From Figure 6, we observe more than 60% reduction in handover latencies, especially in topologies with lower processing power at edge nodes. We find that the number of packet transfers and average handover latency are correlated because handover latencies are mainly due to packet transfers in the network, but vary slightly due to difference in link costs.
- *Overall cost*: This is computed as the sum of the processing cost (Eq. O2-1), the intra-traffic class communication cost (Eq. O2-2), and the inter-traffic class communication cost (Eq. O2-3). Ho and Opt reduce the number of packet transfers between processing nodes, hence the communication cost for handovers as well. But apart from handovers, there are other kinds of requests as well, which use the same VNFs. It is interesting to observe the differences in operational cost when using the proposed approaches as compared to approaches like $MinCost$, whose primary aim is to reduce the overall cost. Ho is designed to minimize the handover latency without considering the total cost, and Opt is designed to minimize the handover latency while minimizing the overall cost. So Opt has low operational cost, as observed in Figure 7. $Greedy$ approach tries to place VNFs at edges as much as possible. Edges nodes are usually costlier in terms of processing cost, hence this approach does not perform well as compared to the other approach.

  Although $MinCost$ approach has been designed to minimize the overall cost, it still costs more than Opt for some cases. This is because each VNF in $MinCost$ approach requires larger processing power to be available at single location as compared to VNF of Opt approach. Hence, it is possible that $MinCost$ approach may fit the VNFs in costlier processing nodes due to the unavailability of required processing power, leading to increase in overall operational cost. However VNFs of Opt require lesser processing resource and can be placed on those low cost locations, thus provides a low cost solution.

We also computed time taken by Opt on a graph, and the sum of the time taken in solving for subgraphs in $SubGr$ approach. Figure 8 shows that $SubGr$ is highly scalable, and takes 275 CPU seconds time to solve for a 320 nodes topology, where as Opt takes 714 CPU seconds to solve for 32 nodes topology. It should be noted that these values for the subgraph based approach are observed by solving all subgraphs sequentially. If solved in parallel, it'll further reduce the solving time. In summary, we observe that our approaches Opt and $SubGr$ have comparable operational cost to the $MinCost$ approach but reduce the average time to serve handover requests by upto 60%. Also, $SubGr$ approach is scalable, with no impact on the optimality of solutions. The results are similar for traffic mix with 10, 15 and 20 percentage handover requests. So, our conclusions are not sensitive to the percentage of handover requests in the network.

VI. Conclusion

The flexibility of deploying Virtual Network Functions in the network quickly to cater to the varying demands of users makes it suitable for emerging 4G-Advanced and 5G telecommunication networks. In this work, we discussed a mobility-aware VNF placement problem and proposed an optimization technique that aims at minimizing the average latency in serving handover requests. We also proposed an improved model that minimizes the average handover latency while reducing the overall operational cost as much as possible. We compared our approach with two approaches and showed 60% reduction in the average handover latency with no considerable increase in the overall operational cost. We also proposed a divide-and-conquer subgraph-based approach to reduce the time taken to obtain placement for larger topologies where integer program could not be solved in reasonable time. We found that the subgraph-based approach can solve the optimization problem for 360 node topology in lesser time than that taken by our previous approach for 28 nodes. Implementing approaches like ours to place VNFs in scenarios with high mobility can improve user experience through quicker handovers.

TABLE II
EVALUATION SETUP PARAMETERS

Parameter	Value
Range of core node cost ($/GHz)	[0.05, 0.10]
Range of edge node cost ($/GHz)	[0.20, 0.25]
Core to core link cost ($/Mbps)	0.005
Core to edge link cost ($/Mbps)	0.015
Edge to edge link cost ($/Mbps)	0.025
Latency between core nodes (sec)	0.015
Latency between core to edge nodes (sec)	0.010
Latency between edge nodes (sec)	0.005
Number of edge node per core node	3
Processing power at core nodes as perc. of processing power reqd by users connected to edges of that core node	80
Probability of links between edge nodes connected to same core node	0.2
Number of core gateway nodes (as %age of core nodes)	75
Range of number of base stations per edge node	[30, 50]
Range of number of active users per base station	[500, 1500]
Frequency of requests per user (/sec)	1
Number of mobility classes	2
Percentage of static users (mobility class = 1) and mobile users (mobility class = 2)	70, 30
Percentage of initial attach and data transfer requests by static users	10, 90
Percentage of initial attach, data transfer, and handover requests by mobile users	10, 85, 5

REFERENCES

[1] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, “Near optimal placement of virtual network functions,” in *Proc. of IEEE INFOCOM ’15*.

[2] A. Baumgartner, V. S. Reddy, and T. Bauschert, “Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization,” in *Proc. of IEEE NetSoft ’15*.

[3] B. A. Huberman and P. Sharma, “Compare: Comparative advantage driven resource allocation for virtual network functions,” in *Proc. of IEEE NFV-SDN ’16*.

[4] M. Bagaa, T. Taleb, and A. Ksentini, “Service-aware network function placement for efficient traffic handling in carrier cloud,” in *Proc. of IEEE WCNC ’14*.

[5] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, “Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions,” in *Proc. of IFIP/IEEE Symposium of Integrated Network Management (IM) ’15*.

[6] D. Krishnaswamy, R. Kothari, and V. Gabale, “Latency and policy aware hierarchical partitioning for nfv systems,” in *Proc. of IEEE NFV-SDN ’15*.

[7] J. G. Herrera and J. F. Botero, “Resource allocation in nfv: A comprehensive survey,” *Proc. of Transactions on Network and Service Management ’16*, vol. 13, no. 3, pp. 518–532.

[8] M. Zekri, B. Jouaber, and D. Zeghlache, “A review on mobility management and vertical handover solutions over heterogeneous wireless networks,” *Computer Communications ’12*, vol. 35, no. 17, pp. 2055–2068.

[9] P. Bertin, S. Bonjour, and J.-M. Bonnin, “A distributed dynamic mobility management scheme designed for flat ip architectures,” in *Proc. of NTMS ’08*.

[10] D. Stynes, K. N. Brown, and C. J. Sreenan, “Using opportunistic caching to improve the efficiency of handover in lte with a pon access network backhaul,” in *Proc. of IEEE LANMAN ’14*.

[11] M. Tawarmalani and N. V. Sahinidis, “A polyhedral branch-and-cut approach to global optimization,” *Mathematical Programming ’05*, vol. 103, no. 2, pp. 225–249.

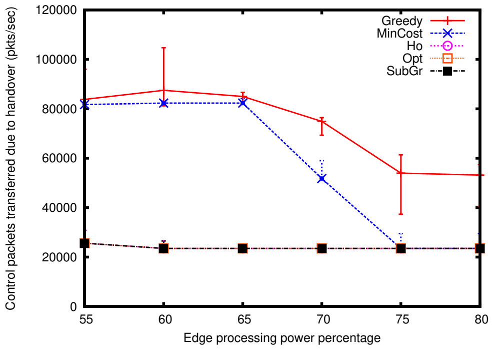

Fig. 5. Comparison of the number of packets due to handover per unit time for different approaches (min, median, and max values have been plotted)

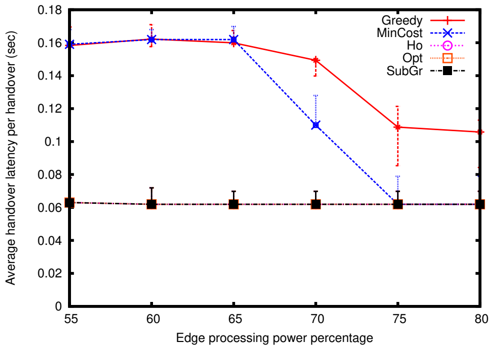

Fig. 6. Comparison of the average handover latency for different approaches (min, median, and max values have been plotted)

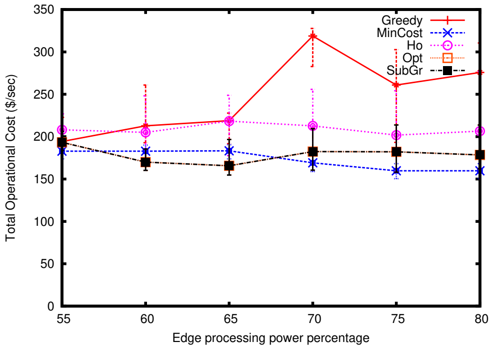

Fig. 7. Comparison of the total operational cost per unit time between different approaches (min, median, and max values have been plotted)

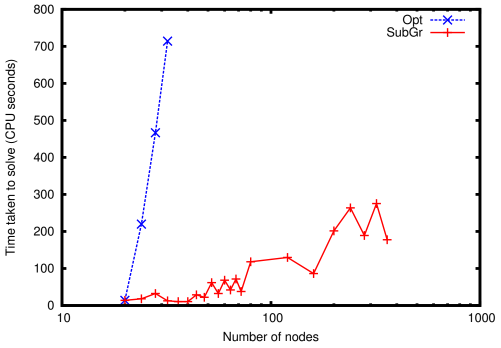

Fig. 8. Comparison of time taken to obtain the optimal placement by running model on complete graph vs subgraphs