# Virtual Function Placement and Traffic Steering over 5G Multi-Technology Networks

Nabeel Akhtar*, Ibrahim Matta*, Ali Raza*, Leonardo Goratti†, Torsten Braun‡ and Flavio Esposito§
*Boston University, USA †FBK CREATE-NET, Italy ‡University of Bern, Switzerland §Saint Louis University, USA
{nabeel, matta, araza}@bu.edu, lgoratti@fbk.eu, braun@iam.unibe.ch and espositof@slu.edu

***Abstract*—Next-generation mobile networks (5G and beyond) are expected to provide higher data rates and ultra-low latency in support of demanding applications, such as virtual and augmented reality, robots and drones, etc. To meet these stringent requirements, edge computing constitutes a central piece of the solution architecture wherein functional components of an application can be deployed over the edge network so as to reduce bandwidth demand over the core network while providing ultra-low latency communication to users. In this paper, we investigate the joint optimal placement of virtual service chains consisting of virtual application functions (components) and the steering of traffic through them, over a 5G multi-technology edge network model consisting of both Ethernet and mmWave links. This problem is NP-hard. We provide a comprehensive "microscopic" binary integer program to model the system, along with a heuristic that is one order of magnitude faster than solving the corresponding binary integer program. Extensive evaluations demonstrate the benefits of managing virtual service chains (by distributing them over the edge network) compared to a baseline "middlebox" approach in terms of overall admissible virtual capacity. We observe significant gains when deploying mmWave links that complement the Ethernet physical infrastructure. Moreover, most of the gains are attributed to only 30% of these mmWave links.**

## I. Introduction

Next-generation mobile networks (5G and beyond) are expected to go beyond the delivery of static or streaming content, such as telephony, web browsing, and low-resolution video. They should be capable of serving many billions of users and smart devices at much higher data rates (over 500 Mbps) and ultra-low latencies (less than 5 milli-seconds) [1], [2], [3]. Potential 5G applications include robots and drones, virtual and augmented reality, healthcare, etc. Traditional network and application architectures can not support these stringent application requirements. Advances in the physical network infrastructure, *e.g.*, the integration of Gigabit Ethernet and mmWave technologies, and the virtualization of network and application functions are key to achieving these 5G goals [1], [2], [3].

The virtualization of network functions, termed Network Function Virtualization (NFV), aims to decouple network software from proprietary, dedicated hardware appliances, termed "middleboxes" (*e.g.*, traffic shapers, Network Address Translation boxes). Similarly, application virtualization allows an application to work in an isolated virtualized environment. Moreover, in cloud-based or service-oriented application architectures, an application can be composed of many application components, where each component can run as a Virtual Function (VF). Thus, under application service virtualization, multiple VFs can run on any general-purpose computer within a virtual machine, in an operating system container, or as a serverless "Function as a Service" (FaaS). The flexibility with which VFs can be deployed and managed — *i.e.*, chained, allocated resources, migrated — allows their hosting "close" to the users, in an edge cloud / datacenter, thus meeting the 5G application requirements of ultra-low latency and high throughput.

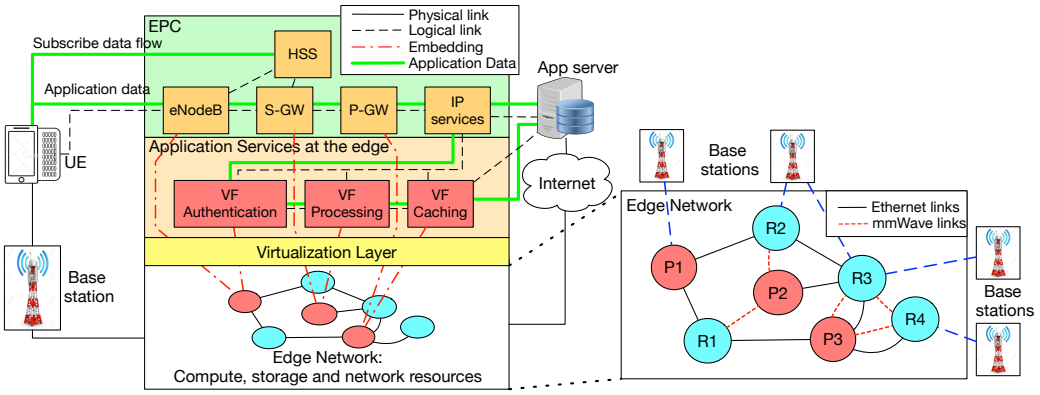
978-1-5386-4633-5/18/$31.00 © 2018 IEEE

Figure 1a illustrates the evolution of cellular networks to 5G, where network services are moved from radio base stations and gateways into the edge cloud. In a traditional LTE architecture, user traffic traverses a series of devices on its way to the application server: the base station (eNodeB), a serving gateway (S-GW), and finally a packet data network gateway (P-GW) that connects to the outside world. On the other hand, in a virtualized environment, these network functions are envisioned to run virtualized, anywhere on the edge resources. They are chained together in a particular order based on processing requirements — in Figure 1a example, (eNode, S-GW, P-GW). To steer traffic across these VFs, Software Defined Networking (SDN) mechanisms are leveraged so that routes are established programmatically between components of the service chain.

Applications running on the edge network can also have different service chain requirements (*e.g.*, Authentication, Processing, Caching in Figure 1), and multiple application flows may need to use the same VF. Thus, understanding where to place VFs, or instances of the same VF, that are necessary to satisfy service chain requirements of different application flows, subject to physical resource (host and network) constraints, is a challenging problem. Furthermore, a 5G edge network may consist of multiple link technologies, *e.g.*, Ethernet and mmWave, that may have different characteristics suitable for possibly different types of application flows.

**Our Contribution:** In this paper, leveraging optimization theory, we investigate the joint placement of virtual service chains consisting of virtual application functions (components) and the steering of traffic through them, over a 5G multi-technology edge network model consisting of both Ethernet and mmWave links. Our contributions are:

- We propose a detailed "microscopic" binary integer program (BIP) to find the optimal placement of virtual functions.
- BIP is NP-hard (*i.e.*, computationally expensive), so we provide a heuristic that is one order of magnitude faster than BIP.
- Our workload model captures virtual service chains that correspond to the needs of 5G applications described as "killer applications" (*i.e.*, virtual and augmented reality) over the edge network.




a) 5G Infrastructure supporting virtualized services

b) Edge network with processing & routing nodes and heterogeneous transport network

Figure 1: Function virtualization for 5G mobile network

- Extensive evaluation results demonstrate the benefits of managing virtual service chains (by distributing them over the edge network) compared to a baseline "middlebox" approach (where all functions are run on one host).
- We observe significant gains when deploying mmWave links that complement the Ethernet physical infrastructure. Moreover, most of the gains are attributed to only 30% of these mmWave links, which indicates that judicial placement of mmWave links is key for maximum gains.
- To the best of our knowledge, this is the first work to study a multi-technology based edge infrastructure envisioned for 5G networks. The developed model can be used by a 5G "service" provider to optimally allocate resources to service chains, and by a 5G "infrastructure" provider to understand the benefits of deploying mmWave links.

**Paper Organization:** The paper is organized as follows: Section II provides a background and reviews related work. Section III describes our system model. Section IV explains our mathematical formulation. Section V presents our evaluation model, parameters and proposed heuristic. Results are shown in Section VI. Section VII concludes the paper.

## II. Background and Related Work

This section provides a review of some of the most prominent research work on function placement and traffic steering, and the industry's direction to support high data rate and ultra-low latency applications on next-generation mobile networks (5G and beyond). According to "IMT-2020", a program developed by the International Telecommunication Union's Radiocommunication Sector (ITU-R) for 5G, the peak data rates are expected to be around 10 Gbits/s, while end-to-end latency is expected to be less than 5 ms [4]. To meet these strict requirements, there is a need for changes in the infrastructure (*e.g.*, using millimeter wave) and for having elasticity in hosting VFs at the edge of the network. Users accessing application servers hosted in the public network experience average delays of 50-100ms, while such applications hosted in the operator's cloud experience delays ranging from 20-50ms. However, these delays are still significantly higher than those expected from a 5G network. To meet the strict requirements of 5G network applications for delays of 1-5 ms, the edge computing paradigm that places computation closer to end users is necessary [1], [2]. As an example, *Telefonica*, one of the world's largest telecom operator, is using their central offices (COs) as datacenters (COdc). These COdc are closer to the end users (at the network edge) and are capable of hosting user VFs [5].

Figure 1a shows the case where service-chain components are running as virtualized functions at the edge of the network. Here, all the traffic from users passes through Authentication, Processing, and Caching, which are running at the edge of the network, before arriving at the Application Server. Note that the operator's network services (*e.g.*, S-GW and P-GW), which are part of the Evolved Packet Core (EPC), can also be virtualized and hosted in the edge datacenter, as shown in Figure 1a. However, in this work, we are specifically studying virtual functions for applications running on the 5G network. *The internal functional split of the 5G RAN and virtual EPCs is beyond the scope of this work.*

Figure 1b shows an example of an edge network, consisting of processing nodes (P1-P3) and routing / switching nodes (R1-R4). This edge network covers a small geographical area, *e.g.*, a medium-size city. As the name suggests, processing nodes have processing power and can host VFs, while routing nodes are responsible for routing traffic through the network. Note that a processing node can also act as a routing node. All the nodes are SDN enabled and can be programmed for traffic routing. The nodes are connected with two different link technologies, namely Gigabit Ethernet and millimeter wave (mmWave) links. The mmWave technology is considered an important aspect of 5G networks. The enormous amount of spectrum available in the mmWave band, and the ease and flexibility of deploying mmWave infrastructure, will greatly increase the network capacity, as well as decrease latency when mmWave links are used to create shortcuts between nodes [6].

Application service components can be hosted at processing nodes. These components run as VFs and can be dynamically instantiated, migrated or removed from the network based on the system requirements. Applications can have strict requirements for their traffic to traverse virtualized services in a certain order, *e.g.*, authentication followed by caching. This is known as "Service Function Chaining" (SFC). SFC is an important capability of virtualized networks as it provides both modularity and elasticity. A single function in a service chain can be dynamically changed/updated without having any impact on other functions. The efficient placement of

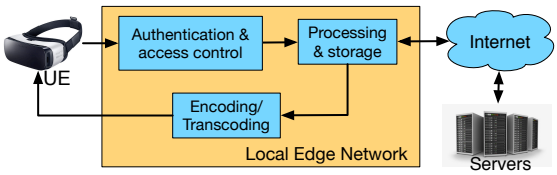



Figure 2: Virtual Reality use case

virtualized functions and traffic steering through service chains are challenging problems.

*1) Placement:* Placement of functions deals with the efficient instantiation of virtual function (VF) instances on processing nodes to satisfy the demands of the system while minimizing the overall cost. Since different application flows can have different service chain requirements, a virtual service graph, with resource requirements, is created for each flow. This graph is embedded on a virtualized physical infrastructure, as shown in Figure 1a. The task of creating and deploying virtual service chains is similar to the Virtual Network Embedding (VNE) problem [7], [8]. Similar to VNE, this task is NP-hard. Different VF placement schemes have been proposed [9], [10]. Formulated as an optimization problem, VF placement and chaining reduces to an integer program, which is NP-hard and intractable for larger inputs. Hence, most solutions focus on designing heuristic or meta-heuristic algorithms for solving the VF placement with service chaining [9], [11]. These solutions aim to quickly find a sub-optimal placement, and are based on simple cost functions and constraints. In this work, *we aim to find an optimal placement based on a detailed system model that captures many complexities that arise with virtualized services for a 5G network, including multi-technology links and detailed service demands. Moreover, we provide a heuristic solution to quickly solve the problem while sacrificing little on the quality of the results.*

*2) Traffic Steering:* Traffic steering through VF instances residing at different locations brings a different set of challenges. Traditionally, traffic is directed through a desired sequence of network functions (middleboxes) using manual configurations, which cannot be imported to the VF paradigm. Since resources are dynamically allocated, there is a need for autonomic traffic steering. SDN offers a flexible control approach and enables traffic forwarding. However, SDN capabilities have been limited to L2/L3 forwarding functions and cannot support VFs. SDN based solutions have been proposed [12], [13], [14], which extend the current L2/L3 functions of SDN to provide a policy enforcement layer for VF traffic steering. Although extended SDN mechanisms have enabled VF traffic steering, finding the best path through a set of VFs under multiple constraints is NP-hard [15]. Previous work focuses on finding paths given the cost function of a single link technology [9], [16]. In this paper, *we consider multiple link technologies, each has its own cost definition. Furthermore, the link cost function takes into account multiple cost metrics to accurately model the link technologies.*

## III. System Model

This section describes our envisioned 5G system model for edge computing. We also describe our use cases (augmented and virtual reality applications) which have stringent processing and communication requirements that “thin" clients / mobile devices and traditional networks fail to support.

Our model of the 5G infrastructure consists of a multi-technology edge network, where nodes are connected with mmWave and Gigabit Ethernet links, as shown in Figure 1b. Nodes that are closer than a threshold distance are connected with mmWave links. There are two types of nodes in the network. Routing Nodes (RN) are OpenFlow enabled routers that forward packets to the next hop toward their destination. Processing Nodes (PN) are RNs with processing power, so a PN can also host Virtual Functions (VFs). A PN has multiple processing cores. For simplicity, we assume that a single core can only host a single VF instance.

There are costs associated with using the network. There is a fixed cost of running a VF instance on a PN. There are two different types of cost associated with using a communication link, namely, fixed cost and usage cost. A fixed cost is incurred if the link is being used, regardless of the amount of traffic flowing through the link. A usage cost is based on the cost per unit of traffic flowing through the link.

Each flow in the network has a source node, destination node, capacity demand, delay demand, and service chain. The capacity demand is the bit rate that a flow needs on each link as it goes from its source to destination. The delay demand is the maximum delay that packets of the flow can experience as they move from the source to destination. A service chain, as we discussed earlier, is an ordered list of VFs that the flow should pass through before reaching the destination node. This is shown in Figure 1a where an application flow passes through VFs running *Authentication*, *Processing*, and *Caching* before reaching the destination application server.

*Online vs Offline:* The resource allocation problem consists of placement of VFs and traffic steering, and it can be done either *online* or *offline*. In the *online* case, the resources are dynamically allocated for each flow as the flow arrives to the system. In the *offline* case, all the flow demands are known in advance and the resources are simultaneously allocated for all flows. Both the *online* and *offline* cases are NP-hard [17]. The *offline* resource provisioning case is not always possible, especially when users’ behavior cannot be accurately predicted. In this paper, we only consider the *online* case. In the next section, we provide a detailed Binary Integer Programming (BIP) formulation for this problem, which can be used for both *online* and *offline* cases. Note that the *online* case is merely the *offline* case with a single flow.

To evaluate our system, we model the workload of service chains inspired by applications such as augmented and virtual reality applications. These applications have stringent requirements, and are described as “killer applications" for the 5G network [1], [2]. For example, VR applications requires high throughput and ultra-low latency. It is believed that VR applications, where users interact with other users, would need bandwidth up to 500Mbps and latency less than 5ms [1]. The challenge in advancing and deploying such applications is that traditional architectures (using remote clouds/datacenters) fail to satisfy such stringent requirements. To overcome this challenge, the VR application should be refactored as a chain of VFs that get deployed at the edge cloud. For instance, the

3D distributed game described in [18] may be decomposed into a chain of VFs as illustrated in Figure 2. The aim is to move most computation from Application Servers to the 5G edge network, to reduce latency and increase throughput. As shown in Figure 2, when a user's request arrives, it first goes through the *Authentication and Access Control* VF to identify the user and check if the user is allowed to make the request. The request then moves to the *Processing and Storage* VF where the request is processed and actions are taken. These actions are also propagated to application servers over the Internet to update the global state of the game. This VF also has storage capability so it can provide caching and deliver data directly to the user. The delivered data finally moves through the *Encoding/Transcoding* VF, where data is encoded/transcoded before being sent to the user.

## IV. Mathematical Model

In this section, we present the Binary Integer Programming (BIP) formulation for the joint placement of virtualized services (VFs) and traffic steering across the service chains. Although our formulation targets our envisioned 5G system model described in Section III, it can be applied to other scenarios by making appropriate changes to cost functions or constraints. Our model can be used by a 5G "service" provider to optimally allocate resources to service chains, as we describe in this section. Specifically, we aim to minimize the operational (OPEX) cost by maximizing the resource usage of the physical infrastructure. All network parameters are described in Table I. (Later in Section V, we use this model, in conjunction with a network graph generation model, to also understand the benefits of deploying mmWave links from the point of view of a 5G "infrastructure" provider.)

| Notation | Description |
|---|---|
| $G(V, E)$ | Network graph, $V$ is the set of nodes: Routing Nodes (RNs) and Processing Nodes (PNs), and $E$ is the set of all links $(u, v)$. |
| $w_{(u,v)}$ | binary {0,1}: 1 if there exists a physical link between nodes $u$ and $v$, 0 otherwise. |
| $c(u, v)$ | Capacity of link $(u, v)$. |
| $l(u, v)$ | Latency of link $(u, v)$. |
| $k^c_{(u,v)}$ | Fixed cost of using link $(u, v)$. If any amount of traffic, greater than zero, passes through link $(u, v)$, we incur this cost. |
| $k^d_{(u,v)}$ | Usage cost of using link $(u, v)$. It is the cost of unit flow that passes through link $(u, v)$. |
| $h^n_i$ | Fixed cost of instantiating a VF instance of type $n$ on node $i \in V$. |
| $O_v$ | Set of cores available at node $v \in V$. Each core can support one VF. |
| $U_s$ | Load (in Mbps) that can be served by a single VF $s \in S$. |
| $M^n_i$ | binary {0,1}: 1 if VF $n \in S$ can be supported at node $i$, 0 otherwise. |
| $\phi_s$ | Ratio of outgoing to incoming flow rate through VF $s \in S$. |

Table I: Network Parameters.

In our model, a physical (or logical) network $G(V, E)$ is made up of nodes $V$, and links $E$ between the nodes. Each link has capacity $c(u, v)$ and latency $l(u, v)$. There is a fixed cost $k^c_{(u,v)}$ of using a link, which is the cost incurred if the link is used. There is also usage cost $k^d_{(u,v)}$ of each link, which represents the cost per unit of flow that passes through the link. There is a cost $h^n_i$ of starting a new virtualized function instance on a Processing Node (PN). Each PN has a set of cores available $O_v$, and a single virtual function can run on a single core. There is a load limit $U_s$ (in Mbps) on the load (bit rate) that can be served by the VF instance. A PN $i \in V$ can only host certain types of VF $n$, as indicated by $M^n_i$. The volume of incoming flow and outgoing flow through a VF can be different, *e.g.*, an encryption VF encrypts incoming traffic, so the amount of outgoing traffic leaving the VF is more than the incoming traffic. The ratio of the outgoing bit rate (in Mbps) over the incoming bit rate (in Mbps) for a VF is given by $\phi_s$.

| Notation | Description |
|---|---|
| $F$ | Set of all flows in the network. |
| $s^f$ | Start node of flow $f \in F$. |
| $t^f$ | Destination node of flow $f \in F$. |
| $d^f$ | Initial capacity demand of flow $f \in F$. |
| $l^f$ | Latency demand of flow $f \in F$. Maximum delay that a flow $f \in F$ can tolerate on the path from source to destination. |
| $K$ | Set of all different VFs that can be placed on nodes. |
| $C^f$ | Service chain of flow $f \in F$. Set of VFs that flow $f \in F$ needs to traverse in a specific order, *i.e.* $n_1 \rightarrow n_2 \rightarrow ... \rightarrow n_l$, where $n_i \in K$. |
| $C^f_{st}$ | $C^f_{st} = [n_{sf} \rightarrow C^f \rightarrow n_{tf}]$. The service chain of flow $f \in F$ which includes $s^f$ and $t^f$ nodes. To ensure that the flow starts at node $s^f$ and ends at node $t^f$, two imaginary VFs $n_{sf}$ and $n_{tf}$ are introduced at $s^f$ and $t^f$ nodes, respectively. Since VFs $n_{sf}$ and $n_{tf}$ are only present at $s^f$ and $t^f$ nodes, these nodes are selected as the start and end nodes on the flow's path. |
| $d^{f(m\rightarrow n)}$ | Capacity demand of flow $f \in F$ from VF $m$ to $n$. $d^{f(m\rightarrow n)} = d^f \prod_{i=s^f}^{m} \phi_i$, (note : $\phi_{sf} = 1$) |

Table II: Traffic Parameters.

Table II shows the traffic parameters. Each flow $f$ in the network has a start node $s^f$, destination node $t^f$, initial capacity demand (in Mbps) $d^f$, latency demand (in milliseconds) $l^f$, and a service chain $C^f$. A flow is unsatisfied if any of its constraints is not met. As the flow traverses through the VFs in its service chain, its capacity demand changes based on the VF's $\phi_i$. The capacity demand of a flow between two VFs is given by $d^{f(m\rightarrow n)}$.

### A. Variables

Table III describes our model variables in detail. This includes decision variables, and derived variables (*i.e.*, variables dependent on decision variables).

### B. BIP Formulation

*1) Objective Function:* Our objective is to find the optimal placement of VFs that minimizes the resource fragmentation in the system, *i.e.*, maximizes the utilization of resources. Since physical resources in the network are usually leased or rented from third parties, we aim to maximize the utilization of resources that are already in use as long as we can satisfy the flow demands. Following are the costs that we consider and we aim to minimize.

*VF Deployment Cost:* To run a VF on a node, we assume a pricing / cost model that is similar to Amazon EC2 "dedicated host", in which a fixed cost is paid for leasing / renting the node on which the VF instance is run.

$$\mathbb{V}_c = \sum_{i \in V} \sum_{n \in K} \sum_{a \in O_i} h^n_i X^n_{ia} \tag{3}$$

*Link Fixed Cost:* If a link is used (in any direction) by any of the flows, regardless of the flow demand, we pay a fixed cost. Different link technologies (namely, Ethernet and mmWave links) can have different fixed costs, which we explain in detail later in Section V.

$$\mathbb{E}_c = \sum_{(u,v) \in E} k^c_{(u,v)} x_{(u,v)} \tag{4}$$

| Variables | Description |
|---|---|
| $x_{(u,v)}^{f(m\to n)}$ | binary {0,1}: 1 if link $(u,v)$ is used to reach from VF $m$ to $n$ in the service chain $C^f$ of flow $f \in F$, and 0 otherwise. |
| $x_{(u,v)}$ | binary {0,1}: 1 if any flow uses link $(u,v)$, and 0 otherwise. Note that it is not a decision variable, as it can be derived from $x_{(u,v)}^{f(m\to n)}$. $x_{(u,v)} = 1$ if $$\sum_{f\in F}\sum_{(m\to n)\in C_{st}^f} x_{(u,v)}^{f(m\to n)} + \sum_{f\in F}\sum_{(m\to n)\in C_{st}^f} x_{(v,u)}^{f(m\to n)} > 0 \quad (1)$$ and 0 otherwise. Equation (1) above can also be written as a set of linear constraints as shown below. $$x_{(u,v)} \le \sum_{f\in F}\sum_{(m\to n)\in C_{st}^f} x_{(u,v)}^{f(m\to n)} + \sum_{f\in F}\sum_{(m\to n)\in C_{st}^f} x_{(v,u)}^{f(m\to n)}$$ $$x_{(u,v)} \ge x_{(u,v)}^{f(m\to n)} \quad \forall f \in F,\ \forall (m\to n) \in C_{st}^f$$ $$x_{(u,v)} \ge x_{(v,u)}^{f(m\to n)} \quad \forall f \in F,\ \forall (m\to n) \in C_{st}^f$$ |
| $S_{ia}^{fn}$ | binary {0,1}: 1 if VF $n \in C_{st}^f$ is placed at core $a$ of node $i$ for flow $f \in F$, and 0 otherwise. |
| $X_{ia}^{n}$ | binary {0,1}: 1 if any VF $n \in K$ is placed on core $a$ of node $i$, 0 otherwise. Note that it is not a decision variable as it can be derived from $S_{ia}^{fn}$. $X_{ia}^n = 1$ if $$\sum_{f\in F} S_{ia}^{fn} \ge 1 \quad \forall n \in C^f, \forall i \in V, \forall a \in O_i \quad (2)$$ and 0 otherwise. Equation (2) above can also be written as a set of linear constraints as shown below. $$X_{ia}^n \le \sum_{f\in F} S_{ia}^{fn} \quad \forall n \in C^f, \forall i \in V, \forall a \in O_i$$ $$X_{ia}^n \ge S_{ia}^{fn} \quad \forall n \in C^f, \forall i \in V, \forall a \in O_i$$ |

Table III: Variables.

*Link Usage Cost:* This link usage cost is based on the amount of link resources used by flows. It represents the cost per unit of flow going through a link.

$$\mathbb{E}_d = \sum_{(u,v)\in E} k_{(u,v)}^d \sum_{f\in F}\sum_{(m\to n)\in C_{st}^f} x_{(u,v)}^{f(m\to n)} d^{f(m\to n)} \quad (5)$$

Our objective is to minimize the cost of the system and fragmentation of the resources in the system, while satisfying the flow demands. The objective function is given by:

$$minimize(\ \mathbb{V}_c + \mathbb{E}_c + \mathbb{E}_d\ )$$

This cost minimization is subject to the following constraints:

*2) Link Capacity Constraint:*

$$\sum_{f\in F}\sum_{(m\to n)\in C_{st}^f} d^{f(m\to n)} x_{(u,v)}^{f(m\to n)} \le c(u,v) \qquad \forall (u,v) \in E \quad (6)$$

Each link has a capacity limit. Flows passing through a link should not exceed the capacity of the link.

*3) Flow Latency Constraint:*

$$\sum_{(m\to n)\in C_{st}^f}\sum_{(u,v)\in E} l(u,v) x_{(u,v)}^{f(m\to n)} \le l^f \qquad \forall f \in F \quad (7)$$

Each flow has a latency constraint. A flow, moving from source to destination, should not experience latency greater than its (end-to-end) latency requirement. Here we are only considering network delays, *i.e.,* propagation and transmission delays.

*4) Physical Link Constraint:*

$$x_{(u,v)}^{f(m\to n)} \le w_{(u,v)} \qquad (m\to n) \in C_{st}^f \quad (8)$$

A virtual link along the path of a flow should be using one of the existing physical links given by $w(u,v)$.

*5) Flow Constraint:*

$$\sum_{j\in V} x_{(i,j)}^{f(m\to n)} - \sum_{k\in V} x_{(k,i)}^{f(m\to n)} = \sum_{a\in O_i} S_{ia}^{fm} - \sum_{a\in O_i} S_{ia}^{fn} \quad (9)$$

$\forall i \in V,\ \ (m \to n) \in C_{st}^f$, where VF $n$ is after VF $m$ in the service chain $C_{st}^f$.
This constraint ensures that there is a single continuous path between pair of nodes on which VFs $m$ and $n$ are placed.

*6) VF Placement Constraint:*

$$S_{ia}^{fn} \le M_i^n \qquad \forall f \in F, \forall n \in C_{st}^f, \forall i \in V, \forall a \in O_i \quad (10)$$

VF $n \in C_{st}^f$ can only be hosted on nodes that can host VF $n$.

*7) Single VF Node Selection Constraint:*

$$\sum_{i\in V}\sum_{a\in O_i} S_{ia}^{fn} = 1 \qquad \forall n \in C_{st}^f \quad (11)$$

Only a single node is selected to host a VF in the service chain $C_{st}^f$ of flow $f \in F$.

*8) Node Capacity Constraint:*

$$\sum_{n\in K}\sum_{a\in O_i} X_{ia}^n \le |O_i| \quad \forall i \in V \quad (12)$$

Each free core at a node can host a single VF. The number of VFs hosted at a node is limited by the number of cores available at that node.

*9) VF Capacity Constraint:*

$$\sum_{f\in F}\sum_{n\in C^f} d^{f(m\to n)} S_{ia}^{fn} \le U_n \quad \forall i \in V, \forall a \in O_i \quad (13)$$

Each VF at a node has a capacity limit and can only serve flow demands (in Mbps) within that limit.

## V. Evaluation Model, Parameters and Proposed Heuristic

In this section, we present our evaluation model and parameters for both the edge network and the workload of VR and AR service chains. We then provide a description of our proposed heuristic.

### A. Edge Network Graph

We generate a graph representing the 5G edge network using the widely used network graph generator BRITE [19]. We use BRITE's *random node placement* model for placing nodes in a plane, and BRITE's *Waxman* model for interconnecting the nodes probabilistically [20]. The initial graph that we generate represents a base edge network that consists of only Gigabit Ethernet links. We then augment this base graph with mmWave links to obtain three different types of graph, which are described next.

**EthOnly:** This is the initial graph generated by BRITE. It contains only Ethernet *(EthOnly)* links. An example of such graph is shown in Figure 3a.

**Single:** mmWave links are added to the *EthOnly* graph if the distance between any two nodes in the graph is less than

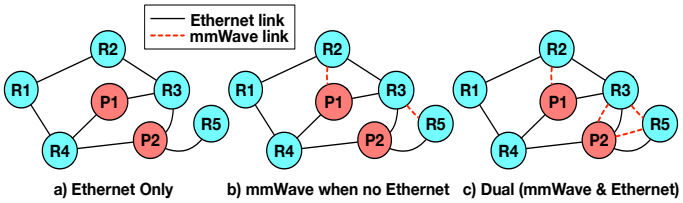



Figure 3: Multi-technology edge network consisting of processing and routing nodes.

| Type | # Nodes | Technology | avg. # of links | %age of links |
|---|---|---|---|---|
| Dual | 25 | mmWave | 47.4 | 65.5 |
| | | Ethernet | 25 | 34.5 |
| Single | 25 | mmWave | 35.6 | 58.8 |
| | | Ethernet | 25 | 41.2 |
| EthOnly | 25 | mmWave | 0 | 0.0 |
| | | Ethernet | 25 | 100 |

Table IV: Graph Parameters and Characteristics

a given distance / mmWave range (elaborated on below). However, if there is already an Ethernet link between the two nodes, a mmWave link is not added. So we have only a *single* type of link technology (mmWave or Ethernet) between any two nodes, as shown in Figure 3b.

**Dual:** mmWave links are added to the *EthOnly* graph if the distance between any two nodes in the graph is less than a given distance / mmWave range (elaborated on below). In this scenario, two nodes may have *dual* technology links, *i.e.*, both mmWave and Ethernet links, as shown in Figure 3c. *Dual* has the maximum number of possible mmWave links between nodes in the network.

We generated different graphs for our evaluation. Characteristics of these graphs, in terms of nodes and links, are summarized in Table IV. We ran our experiments on different graphs of varying densities, but due to lack of space, we only show a representative set of results for five 25-node graphs for each type (*i.e.*, *Dual*, *Single* and *EthOnly*) described above.

Table V shows the various parameters used in our evaluation campaign. The range of a mmWave link is defined by variable $range_{mm}$. Two nodes in the network cannot have a mmWave link if their distance is beyond $range_{mm}$. $range_{mm}$ is chosen to be 500m, which can be achieved in urban environments with LOS [21]. The capacity of mmWave links can vary in the 1 Gbps–10 Gbps range, based on channel conditions [22]. We have taken the link capacity $c(u,v)_{mm}$ to be 2 Gbps for mmWave links [22], and $c(u,v)_{eth}$ to be 10 Gbps for Ethernet.

The fixed cost for using a mmWave link, $k^{c_{mm}}_{(u,v)}$, is kept low by setting it to 1, since it is less costly to establish mmWave links between two sites if they are within the range $range_{mm}$. On the other hand, the fixed cost for Ethernet links is higher, and so we set it to 50, since Ethernet links are usually leased / rented from an infrastructure provider.

The usage cost for mmWave links, $k^{d_{mm}}_{(u,v)}$, is dependent on link performance and is set to $1/PS$, where $PS$ is the probability that a bit sent over the link successfully reaches the other side. $PS$ is obtained using the empirical studies on mmWave technology described in [23], [24]. Figure 4 shows $PS$ as a function of distance. Note that the usage cost $k^{d_{mm}}_{(u,v)}$ becomes significantly higher as the distance between the two nodes connected via a mmWave link increases. Hence, shorter mmWave links are favored over longer mmWave links.

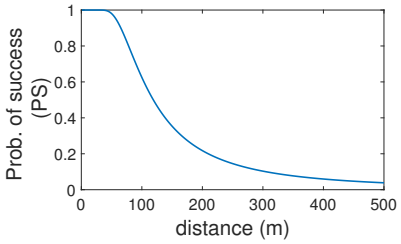



Figure 4: Probability of successful bit delivery over a mmWave link

| Parameter | Description | Value |
|---|---|---|
| $range_{mm}$ | mmWave range | 500 m |
| $c(u,v)_{mm}$ | Capacity of mmWave links | 2 Gbps |
| $c(u,v)_{eth}$ | Capacity of Ethernet links | 10 Gbps |
| $k^{c_{mm}}_{(u,v)}$ | Fixed cost for using mmWave link | 1 |
| $k^{c_{eth}}_{(u,v)}$ | Fixed cost for using Ethernet link | 50 |
| $k^{d_{mm}}_{(u,v)}$ | Cost per unit flow for using mmWave link | $1/PS$ |
| $k^{d_{eth}}_{(u,v)}$ | Cost per unit flow for using Ethernet link | 1 |
| $l_{(u,v)}$ | Latency of link $(u,v)$ is the sum of propagation and transmission delays | - |
| $h^n_i$ | Fixed cost of instantiating a VF instance of type $n$ on node $i$ | 200 |
| $\|O_v\|$ | Number of cores available at processing node $v$ | 4 |
| $U_s$ | Capacity of VF $s$ | 15 Gbps |
| $ratio_{PN}$ | Ratio of processing nodes | 0.3 |

Table V: Evaluation Parameters

For Ethernet links, the usage cost $k^{d_{eth}}_{(u,v)} = 1$, since the cost (delivery performance penalty) associated with using Ethernet is relatively much lower. The latency of a link is given by $l(u,v)$, and is equal to the sum of propagation and transmission delays. Note that there will be zero or negligible queueing delays when demands match allocated capacities.

We randomly select a fraction of the nodes in the network graph to be processing nodes (PNs). This ratio, denoted by $ratio_{PN}$, is set to 0.3, *i.e.* only 30% of the nodes are PNs. Each PN node has $|O_v|$ cores available, and we set $|O_v| = 4$. This means that each PN can host at most 4 VFs. The capacity of a single VF $U_s$ is set to 15 Gbps.

The cost associated with instantiating a VF $h^n_i$ is set to 200. It represents the cost of leasing a virtual machine or container from the edge datacenter. A high value has the effect of packing as many flows as possible on a VF as long as the flow demands can still be fulfilled.

### B. Input Flow Parameters

There are two different types of flow in the network, each type has different service chain requirements representing either Virtual Reality (VR) or Augmented Reality (AR). For each of the generated network graphs, we generate five sets of flows, where each incoming flow is either VR or AR flow with probability 0.5. Each flow starts and ends at the same node (representing the user/client), which is randomly selected. We only consider the allocation of the service chains on the edge network. Flow parameters for VR and AR flows are described in Table VI.

| VR Flow | | |
|---|---|---|
| **Parameter** | **Description** | **Value** |
| $d^f_{VR}$ | Initial flow demand | $\mu$ = 10 Mbps<br>$\sigma$ = 2 Mbps |
| $l^f_{VR}$ | Latency demand | $\mu$ = 5 ms<br>$\sigma$ = 1 ms |
| $\phi_{A\&AC}$ | Ratio of outgoing to incoming flow rate through the Authentication & access control VF | 0.9 |
| $\phi_{P\&S}$ | Ratio of outgoing to incoming flow rate through the Processing & storage VF | 20 |
| $\phi_{E\&T}$ | Ratio of outgoing to incoming flow rate through the Encoding / Transcoding VF | 0.8 |
| **AR Flow** | | |
| **Parameter** | **Description** | **Value** |
| $d^f_{AR}$ | Initial flow demand | $\mu$ = 150 Mbps<br>$\sigma$ = 20 Mbps |
| $l^f_{AR}$ | Latency demand | $\mu$ = 4 ms<br>$\sigma$ = 1 ms |
| $\phi_{A\&AC}$ | Ratio of outgoing to incoming flow rate through the Authentication & access control VF | 0.9 |
| $\phi_{L\&Tk}$ | Ratio of outgoing to incoming flow rate through the Localization / Tracking VF | 0.9 |
| $\phi_{E/P/S}$ | Ratio of outgoing to incoming flow rate through the Embedding / Processing / Storage VF | 1 |
| $\phi_{E\&T}$ | Ratio of outgoing to incoming flow rate through the Encoding / Transcoding VF | 0.8 |

Table VI: Flow Parameters

### C. Proposed Heuristic

We used the CPLEX solver[1] to solve the BIP that we described in Section IV-B. The running time for obtaining the optimal solution for each of our evaluation experiments was up to 50 seconds for 1,000 flow arrivals. To reduce the running time, Algorithm 1 shows a fast heuristic whose solution we compare against the CPLEX solution in terms of performance and running time.

**Algorithm 1** Service Chain Placement Heuristic

**Input:**
$f$: incoming flow
$G(V, E)$: Network graph, $V$ is set of nodes and $E$ is set of links
$PN$: set of processing nodes, where $PN \subseteq V$
$q$: number of nearest processing nodes used for virtual function placement
**Output:** *minPath*

1: $G'$ = *getFeasibleGraph*($G$, $f$); // subgraph G'(V,E'), E' can carry flow demand
2: $PN_q^{s^f}$ = *getNearbyPN*($G'$, $s^f$, $PN$, $q$);// get set of $q$ nearby processing nodes
3: $P^f$ = *getShortestPaths*($PN_q^{s^f}$, $G$, $f$);//all possible paths through processing nodes
4: minPath = *null*
5: minPathCost = $\infty$
6: **for** path $p$ in $P^f$ **do**
7: **if** *pathFeasible*($p$, $l^f$, $d^f$) **then**
8: $c_p^f$ = *getCost*($p$, $f$) // cost = fixed cost + usage cost + VF placement cost
9: **if** $c_p^f$ < minPathCost **then**
10: minPathCost = $c_p^f$
11: minPath = $p$
12: **end if**
13: **end if**
14: **end for**

This heuristic takes a flow and returns a least cost path, while fulfilling the flow requirements. It takes the current state of the network graph ($G$ with nodes, edges, residual link capacities, fixed and dynamic costs, processing nodes) as input, along with the input flow requirements, *i.e.*, source and destination nodes, service chain, flow latency and $\phi_i$ (bandwidth ratio after the use of each VF along the chain).

Initially (line 1), we use the function *getFeasibleGraph* to get a subgraph ($G'$) from the original graph $G$ that includes only those links that have enough capacity to satisfy the flow end-to-end rate demand. *getNearbyPN* (line 2) finds the $q$ nearest (in number of hops) processing nodes ($PN_q^{s^f}$) from the source ($s^f$) on subgraph $G'$ using Dijkstra's shortest path algorithm. After getting $PN_q^{s^f}$ processing nodes, *getShortestPaths* is invoked (line 3) which calculates all possible least-cost paths through every permutation of the processing nodes in $PN_q^{s^f}$. While calculating paths, *getShortestPaths* makes sure that for each path, the segment from the source to the first processing node has available capacity that is at least equal to the rate outgoing from the source ($d^f$). Also, from the first processing node to the last processing node, it has the maximum possible capacity required by the flow, and from the last node to the destination, it has at least a capacity of $d^f \prod_{i=s^f}^{n_l} \phi_i$.

Next, we evaluate each path individually. We perform additional feasibility checks using *pathFeasible* in line 7. *pathFeasible* checks if the path's latency is less than the flow's end-to-end latency requirement and the path can provide/deploy the function chain. If the path is feasible, we calculate the cost of allocating the flow $f$ on the path $p$ using the function *getCost* (line 8); the cost includes link usage and VF deployment cost along the path $p$. Here, we take a greedy approach where we try to use VFs that are already deployed along the path, otherwise collocate other missing VFs on the same processing node(s) if feasible. After evaluating all paths in $P^f$, we pick the path with the lowest cost for the flow.

[1]IBM ILOG CPLEX Optimizer,
http://www-01.ibm.com/software/integration/optimization/cplex-optimizer

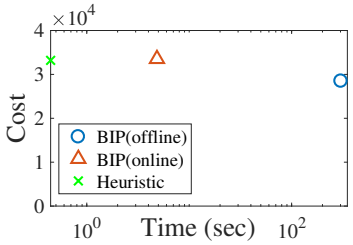
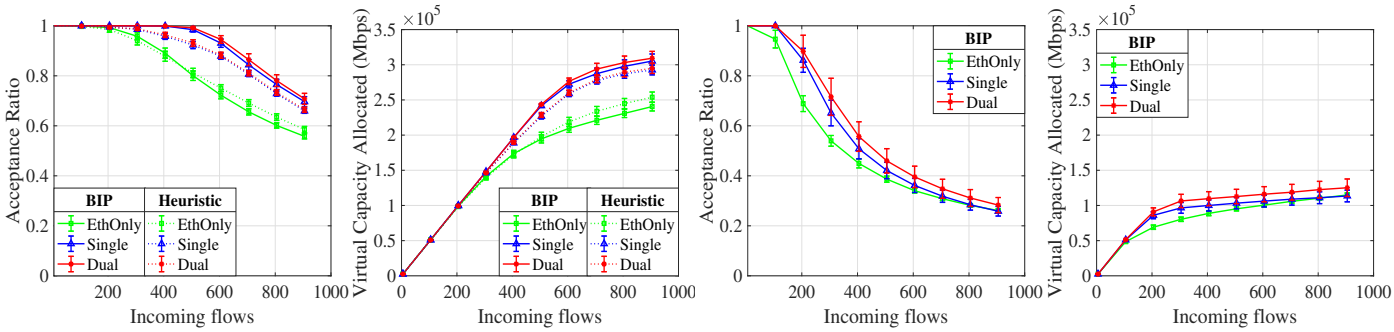



Figure 5: Cost vs running time comparison of BIP vs Heuristic

## VI. Evaluation Results

In this section, we discuss the results of our study where we evaluate the performance and cost of allocating service chains as flows arrive to the 5G edge network. We consider the following performance metrics: (1) *Flow Acceptance Ratio*: is the ratio of flows accepted (*i.e.*, resources are available to allocate to these flows) to the total number of flow arrivals, (2) *Virtual Capacity Allocated*: is the total virtual capacity of all links along the service chains of accepted flows, and (3) *Average Link Utilization*: is the ratio of link usage over link capacity averaged over all links, or over each of the two types of link (Ethernet and mmWave). Results with 90% confidence intervals are shown for *EthOnly*, *Single*, and *Dual* networks for both BIP and Heuristic.

**Observations:** Before presenting the details of our results, we summarize our main observations as follows: (1) Augmenting the physical Ethernet infrastructure with mmWave links yields significantly higher flow acceptance ratio and virtual capacity allocated (up to 20% higher); (2) These mmWave links should complement the connectivity provided by Ethernet and only




(a) Acceptance Ratio (b) Virtual Capacity Allocated (c) Acceptance Ratio for Middlebox scenario (d) Virtual Capacity Allocated for Middlebox scenario

Figure 6: Flow Acceptance Ratio and Virtual Capacity Allocated for *EthOnly*, *Single* and *Dual* networks with BIP and Heuristic, and comparison with the Middlebox approach

a small number of mmWave links needs to be deployed to achieve most performance gains; (3) The flexibility in resource allocation afforded by decomposing 5G applications into service chains that can be deployed *anywhere* on the edge infrastructure yields significant gains (up to three times higher accepted virtual capacity) over a traditional "middlebox" static deployment; and (4) The proposed heuristic decreases the running time by up to one order of magnitude when compared with BIP, while giving performance results very close to BIP.

The cost versus running time for BIP and our heuristic is shown in Figure 5 for the *Dual* scenario. As explained in Section III, in the BIP *online* case, the resources are dynamically allocated for each flow as it arrives, while in the *offline* case, all flow demands are known in advance and resources are simultaneously allocated for all flows. Since *offline* has advance knowledge of all flow demands, it can efficiently allocate the flows on the network and the cost is lowest. However, the running time for *offline* is orders of magnitude larger than the online case. The proposed heuristic yields a cost comparable to the BIP *online* case, with running time that is one order of magnitude lower. The *offline* resource provisioning is not always possible since we cannot accurately predict incoming flows. For this reason, in the remainder of the paper, results are shown for the BIP online case.

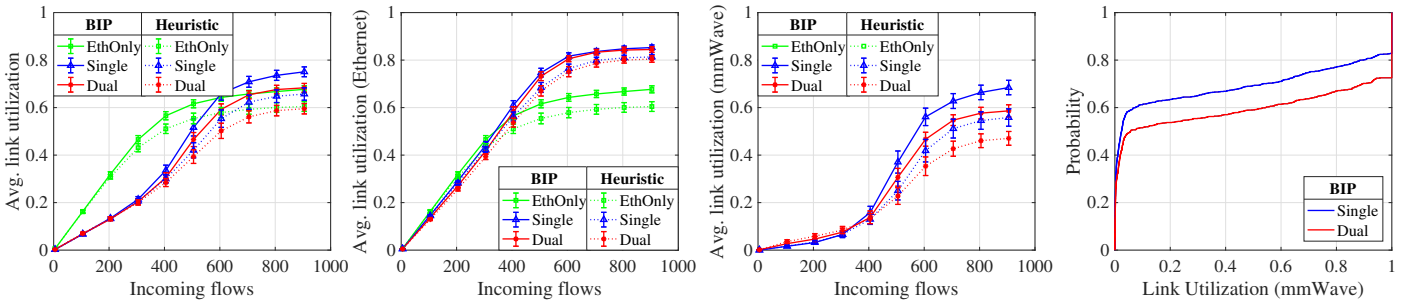
Figure 6a shows the flow acceptance ratio as a function of incoming flows for different types of network. Networks with mmWave links (*Single* and *Dual*) accept more flows than those with only Ethernet links (*EthOnly*). Since each flow can have different capacity requirements along its virtual service chain, the number of flows accepted does not necessarily mean that the network capacity is efficiently allocated. Figure 6b shows the virtual capacity allocated. Again, we see that *Single* and *Dual* have higher virtual capacity allocated than *EthOnly*. For both Figure 6a and Figure 6b, results obtained by the proposed heuristic are very close to BIP.

Figure 7a to 7c shows the average link utilization for both mmWave links and Ethernet. We observe that the *EthOnly* network has higher link utilization because the network has lower capacity and links get congested quickly. Figure 7b and 7c show the link utilization for Ethernet links, and for mmWave links, respectively. We see in Figure 7b that Ethernet links are better utilized (up to 20%) when there are mmWave links in the network. The existence of mmWave links makes the network better connected, which leads to better utilization of the resources and higher number of flows accepted. Figure 7c shows that mmWave links are better utilized (up to 10%) in *Single* networks compared to *Dual* networks, although the acceptance ratio and virtual capacity allocated for both networks are the same. However, mmWave links have higher usage cost. Thus, initially, when the network is not yet congested, only a few mmWave links are used. So initially, the average utilization for mmWave links is low, as shown in Figure 7c. On the other hand, as more flows enter the system and the network becomes congested, more and more mmWave links are used to satisfy the flow demands. This leads to higher utilization of mmWave links, but at a higher cost. In all the graphs, we also provide a comparison with the proposed heuristic. We observe that the heuristic performance is close to the optimal performance given by BIP.

Figure 7d shows the CDF of utilization of the mmWave links for both *Single* and *Dual* scenarios. We observe that in the *Single* scenario, 60% of the links have utilization of less than 6%, and around 20% of the links are completely saturated with utilization close to 100%. This shows that significant performance gains can be achieved by judiciously deploying a small number of mmWave links.

**Middlebox Scenario:** To highlight the benefit of using (optimal) distributed virtual NF placement, we compare it with a traditional middlebox scenario. In the middlebox scenario, a powerful hardware appliance, with all the required services, is placed at the edge of the network. For each network (*i.e.*, *EthOnly*, *Single* and *Dual*), we chose a Processing Node (PN) with the highest node degree to host the middlebox, *i.e.*, node with access to highest network capacity. We set this middlebox to be 10 times more powerful (*i.e.*, it can serve 10 times more flows) than a virtualized service placed on a PN, and it runs all the needed services. Figures 6c and 6d show the flow acceptance ratio and virtual capacity allocated, respectively, for the middlebox scenario. The number of flows accepted in the middlebox case (Figure 6c) are far lower than that accepted in the distributed VF placement scenario (Figure 6a). As shown in Figures 6b and 6d, the virtual capacity allocated for the distributed VF placement scenario is three times higher than the traditional middlebox scenario for higher density networks.

**Discussion:** The results clearly show the benefits of introducing mmWave links in the network. However, it is important to wisely deploy these mmWave links. As shown in Table IV, the *Dual* network has a larger number of mmWave links compared to the *Single* network. However, if we look at the marginal utility of using *Dual* over *Single*, the gains are negligible. The flow acceptance ratio and the virtual capacity




(a) Average Link Utilization for all links

(b) Average Link Utilization for Ethernet links

(c) Average Link Utilization for mmWave links

(d) CDF of utilization of the mmWave links

Figure 7: Average Link Utilization as a function of incoming flows for *EthOnly*, *Single* and *Dual* networks, and the CDF of mmWave link utilization

allocated (Figures 6a and 6b) for both cases are within the 90% confidence interval. Furthermore, the average utilization of links is higher in *Single* compared to *Dual* (Figure 7a), which means links are better utilized in the former. Figure 7d also shows that only a small number of mmWave links are needed to achieve most performance gains. This leads us to conclude that a small number of mmWave links should be introduced such that the overall connectivity between the nodes is increased, rather than to just increase the capacity of the network.

We also note that the middlebox scenario fails to take advantage of introducing mmWave links, as the number of flows accepted for the *EthOnly* network is similar to that for networks with additional mmWave links (Figures 6b and 6d).

## VII. Conclusion

In this paper, we studied the problem of allocating resources at the edge of a 5G network in support of envisioned 5G applications, *e.g.*, virtual and augmented reality. We presented a model of a 5G edge network with multiple link technologies, namely, Ethernet and mmWave. We also developed a workload model that consists of the service chains with varying capacity requirements as the traffic flow traverses its chain. We formulated a binary integer optimization problem whose objective is to minimize the cost of deploying these service chains over the edge network, while satisfying their high throughput and ultra-low latency requirements. We also introduced a fast heuristic to solve the problem. Our extensive evaluations demonstrate the benefits of managing virtual service chains (by distributing them over the edge network) compared to a baseline "middlebox" approach (where all services are run on one host) in terms of overall admissible virtual capacity. Moreover, we observe significant gains when deploying a small number of mmWave links that complement the Ethernet physical infrastructure. We believe this work is a first step toward further analysis and implementation of edge cloud-based 5G applications.

## Acknowledgment

Flavio Esposito's work has been supported in part by the National Science Foundation award CNS-1647084.

## References

[1] AT&T, "Enabling Mobile Augmented and Virtual Reality with 5G Networks," January, 2017. [Online]. Available: goo.gl/DKNbHx

[2] Qualcomm, "Augmented and Virtual Reality: the First Wave of 5G Killer Apps," February 1, 2017. [Online]. Available: https://goo.gl/kdgvCg

[3] Y. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing:A key technology towards 5G," *ETSI W. Paper*, vol. 11, 2015.

[4] "Report ITU-R M.[IMT-2020.TECH PERF REQ] - Minimum requirements related to technical performance for IMT-2020 radio interface(s)," 2017-02-23. [Online]. Available: https://www.itu.int/md/R15-SG05-C-0040/en

[5] R. S. Montero, E. Rojas, A. A. Carrillo, and I. M. Llorente, "Extending the cloud to the network edge," *Computer*, vol. 50, no. 4, April 2017.

[6] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges," *Wireless Networks*, vol. 21, no. 8, pp. 2657–2676, Nov 2015.

[7] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1888–1906, Fourth 2013.

[8] M. G. Rabbani, R. P. Esteves, M. Podlesny, G. Simon, L. Z. Granville, and R. Boutaba, "On tackling virtual data center embedding problem," in *IFIP/IEEE IM 2013)*, pp. 177–184.

[9] L. Guo, J. Pang, and A. Walid, "Dynamic Service Function Chaining in SDN-enabled networks with middleboxes," in *IEEE ICNP*, 2016.

[10] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *IEEE CloudNet*, 2014.

[11] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: network processing as a cloud service," in *SIGCOMM '12, Finland*, 2012, pp. 13–24.

[12] S. K. Fayazbakhsh, V. Sekar, M. Yu, and J. C. Mogul, "FlowTags: Enforcing Network-wide Policies in the Presence of Dynamic Middlebox Actions," in *ACM SIGCOMM - HotSDN*, 2013.

[13] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "SIMPLE-fying Middlebox Policy Enforcement Using SDN," in *ACM SIGCOMM 2013*.

[14] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patneyt, M. Shirazipour, R. Subrahmaniam, C. Truchan, and M. Tatipamula, "StEERING: A software-defined networking for inline service chaining," in *IEEE ICNP*, 2013.

[15] A. Puri and S. Tripakis, *Algorithms for the Multi-constrained Routing Problem*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[16] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *IEEE CloudNet*, 2015.

[17] A. Schrijver, *Theory of Linear and Integer Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1986.

[18] A. Jurgelionis, P. Fechteler, P. Eisert, F. Bellotti, H. David, J. P. Laulajainen, R. Carmichael, V. Poulopoulos, A. Laikari, P. Perälä, A. De Gloria, and C. Bouras, "Platform for distributed 3d gaming," *Int. J. Comput. Games Technol.*, 2009.

[19] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *MASCOTS '01*. IEEE Computer Society, 2001.

[20] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, Dec 1988.

[21] Y. Azar, G. N. Wong, K. Wang, R. Mayzus, J. K. Schulz, H. Zhao, F. Gutierrez, D. Hwang, and T. S. Rappaport, "28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city," in *IEEE ICC)*, June 2013.

[22] M. WeiSS, M. Huchard, A. Stohr, B. Charbonnier, S. Fedderwitz, and D. S. Jager, "60-GHz Photonic Millimeter-Wave Link for Short- to Medium-Range Wireless Transmission Up to 12.5 Gb/s," *Journal of Lightwave Technology*, vol. 26, no. 15, pp. 2424–2429, Aug 2008.

[23] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, March 2014.

[24] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE JSAC*, vol. 33, Oct 2015.