Joint Virtual Machine Embedding and Wireless Data Center Topology Management

Beyza Butun
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
beyza.butun@metu.edu.tr

Ertan Onur
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
eronur@metu.edu.tr

***Abstract*—Data Center Networks (DCNs) have become critical infrastructures with the enormous increase in the amount of data generated. This increase introduces some problems in DCNs such as high cabling complexity, high-level power consumption, low space utilization, and oversubscription. Wireless DCNs (WD-CNs) address these problems by decreasing cabling complexity and power consumption and increasing space utilization thanks to their incorporation of wireless communication technologies. Moreover, with its highly flexible structure, WDCNs efficiently utilize the bandwidth and provide a promising solution to the oversubscription problem. However, WDCNs pose some challenges such as meeting dynamic traffic demands and increasing throughput in the data center. In this work, we propose a WDCN with dynamic topology, and we aim to handle the dynamic traffic demand of virtual machines (i.e., services) and maximize throughput by efficiently embedding virtual machines into physical machines (i.e., servers) and deploying wireless transceivers on top of the racks to establish dynamic wireless communication using unlicensed 60 GHz bands. To maximize throughput in the WDCN, we create a mixed-integer programming (MIP) problem. Due to the complexity of the problem, we also introduce two heuristics named Heuristic for Wireless Link and Service Deployment (HWSD) and Improved HWSD (I-HWSD). These heuristics significantly improve the time it takes to solve the MIP problem with an optimization solver and approach the optimal solution by about 20%. Unlike top-of-rack (ToR)-to-ToR WDCN, which doesn't consider deploying virtual machines, we increase the throughput we can achieve by deploying virtual machines according to the average traffic demands between them.**

***Index Terms*—Wireless data center network; 60 GHz; Wireless communication**

I. Introduction

Data center networks (DCNs) play an integral role that interconnects all the computational, storage, and network resources that data centers provide together. A data center is equipped with switches, routers, firewalls, and servers. Servers are responsible for storing data and providing services (i.e., applications) and files to client devices, so they need to operate without interruption. Hence, it is important to take the capacity of the servers (e.g., CPU and RAM) into consideration while embedding applications to the servers to ensure reliability.

DCNs have become the backbone of the network with the emerging developments in software networks and emerging technologies such as artificial intelligence, fifth-generation, and robotics [1]. Lately, the data generated by these technologies grows enormously, so DCNs play a crucial role in storing and processing a massive amount of data. Today, fat-tree [2] is one of the most used conventional topologies in DCNs. This topology forms a multi-root tree in which a few switches serve as root nodes at the core layer [3]. In conventional DCNs, connections are established via uniform and fixed capacity cables [4]. With the enormous increase in data volume, global traffic becomes quite dense and unbalanced. Fixed-capacity cables and ultra-dense traffic would create a bottleneck in the root nodes at the core layer, causing oversubscription in transmissions between servers in different racks, which reduces overall throughput [3].

To handle the massive amount of data, in conventional DCNs, we need to deploy large numbers of servers and switches, and concerning it, deploy enormous lengths of wires. Adding more network resources and cables to DCNs increases the complexity and the cost of deployment and management, and this complexity creates obstacles to the reconfigurability and scalability of the network. Moreover, large bundles of wires cause inefficient space utilization and inefficient cooling that exacerbates energy consumption [5], [6]. WDCNs eliminate the need for power-hungry, error-prone, and bandwidth-limited switches and offer more energy and bandwidth-efficient DCN thanks to their cooperation with wireless communication technologies [4]. Wireless technologies eliminate the usage of wires and increase space utilization and cooling and energy efficiency [4]. Moreover, with their highly flexible structure, WDCNs efficiently utilize the bandwidth, meet the dynamically changing traffic demands, and provide a promising solution to the oversubscription.

However, WDCNs have some challenges that need to be addressed. Adapting the topology of WDCNs to dynamically changing traffic by establishing dynamic wireless links is an important challenge. Because WDCNs eliminate wires and replace them with wireless equipment, satisfying the reliability requirements of a data center becomes another difficulty due to intolerance of wireless signals to obstructions [7]. In this study, our goal is to design a WDCN and increase its throughput by deploying virtual machines into physical machines and jointly establishing wireless communication between the transceivers deployed on top of the racks. Deploying virtual machines into physical machines considering traffic demands between them helps us to increase traffic demand met. This study

978-3-903176-36-2

uses internationally available 60 GHz unlicensed millimeter-wave bands to establish wireless communication links. 60 GHz provides higher throughput for data centers compared to lower frequencies and has a short communication range which makes it more useful for indoor scenarios. Our main contributions are:

1) We design a WDCN topology considering wireless communication requirements, and we formulate Wireless Link and Service Deployment (WSD) problem to find optimal service deployment and wireless link establishment scenarios in Section III.
2) Because of the complexity of our problem, we introduce a heuristic named Heuristic for Wireless Link and Service Deployment (HWSD). To further improve the solution of HWSD, we also introduce Improved-HWSD (I-HWSD) in Section IV.
3) We present an extensive analysis of our proposed methods in Section V.

There is no study in the literature proposing a joint solution of wireless link establishment and virtual machine deployment to increase throughput in WDCNs.

II. Related Work

The drawbacks of the wired DCNs, such as the fixed cable capacity, inefficient space utilization, and the need for power-hungry switches, lead researchers to research WDCN architectures. The advances and challenges of WDCNs and some proposed architectures are presented [4], [8]. To handle bottlenecks in wired DCNs caused by fixed cable capacity and satisfy dynamic traffic demand, a hybrid DCN architecture is introduced [3]. Hybrid DCN architectures keep cables but establish wireless links to increase the flexibility of the network. The authors of [3] also propose a mechanism that schedules wireless transmissions based on dynamic traffic demands and state that the congestion in the network can be greatly alleviated with the proposed architecture and mechanism.

In addition to hybrid architecture ideas, researchers also propose completely wireless data center architectures. Rommel et al. [1] address wireless technology beyond 5G such as spectrum unification, beamforming, and self-configuring networks indicating them as a potential solution for wireless data center connectivity and discuss how the introduction of wireless connectivity affects the network architecture and data center design. The characterization of 300 GHz channels with optical lenses for wireless rack-to-rack data center communications is clarified and the impact of obstructions such as cables on THz propagation is evaluated [9]. Cao et al. [10] propose serial and parallel multi-objective evolutionary algorithms to solve the topology optimization problem that simultaneously considers the objectives of propagation intensity, interference intensity, and coverage to build a high-speed WDCN topology. Mamun et al. [11] propose a Top-of-Rack (ToR) to ToR WDCN architecture based on 60 GHz establishing direct single-hop links between the ToRs instead of tree topology. Authors of [11] indicate that the ToR-to-ToR WDCN can obtain similar data rates for typical query-based applications and reduces the power consumption compared to the traditional DCNs. S2S-WiDCN [5] proposes a server-to-server WDCN architecture using mmWave links, and an obstruction-aware adaptive routing algorithm to handle obstruction problems in the data center. S2S-WiDCN reduces the power consumption compared to the traditional fat tree DCNs and ToR-to-ToR WDCN [11] and presents comparable flow completion duration and throughput to a traditional fat-tree-based DCN.

Our study is quite different from the studies done in the scope of completely wireless data centers. Although we deploy transceivers on top of the racks and establish wireless links between them to satisfy traffic demands of applications as [11], different from this study and the others, at the same time we deploy virtual machines into physical machines taking into account the average traffic demands between them.

III. Wireless Link and Service Deployment (WSD) Problem

In this work, we aim to meet the dynamic traffic demand of virtual machines and maximize throughput. For this purpose, first, we design a WDCN consider focusing on the maximum possible distance between transceivers on top of the racks to establish a wireless link. Then, we formulate an optimization problem to maximize the throughput of the WDCN.

A. Wireless Data Center Network Topology

The grid in Fig. 1 shows the top view of the deployed data center (DC), and each rectangle and dot in the grid represents the racks and transceivers deployed on top of the racks, respectively. This grid is on the x and y-axes. In the grid, the racks which are laid out in parallel to the x-axis represent the rows in the DC and there is a 1-meter aisle between each row. Each transceiver on top of the racks has a location which is shown in terms of x and y-coordinates. The coordinate y shows on which row the transceiver is deployed, while x shows which rack the transceiver is deployed on this row. The distance of all deployed transceivers from the ground is equal. We assume a rack has typical dimensions, which are 42U in height (1U=44.45mm), 600mm in width, and 1070mm in depth [12]. Hence, the distance between two transceivers deployed on top of the side by side racks is 0.6 meters and deployed on top of the back-to-back racks is 2.07 meters.

To design a WDCN considering 60 GHz channels, first we should find the maximum possible distance between two transceivers on top of the racks to establish a wireless link. The maximum distance has to satisfy $P_r \geq R_s$, where P_r stands for received signal power and R_s stands for receiver sensitivity that is the weakest signal that a receiver can identify and process. We can calculate the received signal power as

$$P_r = P_t G_r G_t \left(\frac{c}{4\pi d_{uv} f} \right)^2, \tag{1.1}$$

where P_t is transmit power assumed to be 100 milliwatts, G_r and G_t are antenna gains for the receiver and the transceiver respectively, both are considered to be 9 dBi [5], c is the speed of light ($3 \times 10^8 m/s$), f stands for carrier frequency of the channel taken as 60.48 GHz, and d_{uv} is the distance in meters

TABLE I: List of symbols

Symbols	Implication	Unit	Default Value
P_t	The transmit power	mW	100
P_r	The received signal power	mW	-
G_r	Antenna gain of the receiver	dBi	9
G_t	Antenna gain of the transmitter	dBi	9
c	The speed of light	m/s	3×10^8
f	The carrier frequency	GHz	60.48
T	The temperature	K	290
N_{figure}	Noise figure	dB	6
N_{floor}	Noise floor	dB	-
N_p	Noise power	W	-
SNR or $\frac{S}{N}$	Signal-to-noise ratio	-	-
$\frac{E_b}{N_0}$	Energy per bit to noise power spectral density ratio	BER	3×10^{-7}
R_s	The receiver sensitivity	dB	-
B	The bandwidth of the channel	GHz	1.632
R	The bit rate	bps	-
V	The global set of the racks	-	-
P	The global set of the physical machines	-	-
S	The global set of the virtual machines	-	-
m_p	The total memory of the physical machine p	GB	256
h_p	Maximum allowed cache size of the physical machine p	MB	20
c_p	Maximum allowed CPU usage of the physical machine p	-	100%
α_s	The memory usage of the virtual machine s	GB	[4, 128]
β_s	The cache memory usage of the virtual machine s	MB	[1, 18]
γ_s	The CPU usage of the virtual machine s	-	[12.5, 50]%
C_{uv}	The capacity of the channel between the transceivers on top of the racks u and v	bps	-
r_{st}	The average traffic demand from the virtual machine s to the virtual machine t	Gbps	[0, 2]
d_{uv}	The distance between the transceivers deployed on the racks u and v	meter	-
δ_{pu}	The binary value indicating whether the physical machine p is deployed in the rack u	-	-
ξ_{sp}	The binary value indicating whether the virtual machine s is embedded into the physical machine p	-	-
X_{uv}	The binary value indicating whether there is a wireless link between two transceivers deployed on top of the racks u and v	-	-

between two transceivers deployed on the racks u and v, and calculated as

$$d_{uv} = \sqrt{(0.6(x_u - x_v))^2 + (2.07(y_u - y_v))^2} \quad \forall u, v \in V,$$

where (x_u, y_u) and (x_v, y_v) are the locations of u and v, respectively, and V is the global set of the racks. The receiver sensitivity R_s is calculated as

$$R_{s dB} = N_{floor_{dB}} + SNR_{dB}, \tag{1.2}$$

where the N_{floor} and SNR represents noise floor and signal-to-noise ratio, respectively. $N_{floor_{dB}} = N_{p dB} + N_{figure_{dB}}$, where N_{figure} is the noise figure assumed to be 6 dB [13] and N_p is noise power such that $N_p = kTB$, where k is the Boltzmann's constant $(1.38 \times 10^{-23}\ Joules/Kelvin)$, T is the room temperature which is typically 290K, and B is the bandwidth of the channel, which is 1.632 GHz [14]. Moreover, $SNR = \frac{E_b R}{N_0 B}$ or $SNR = \frac{P_r}{kTB}$, where R stands for bit rate in bps, and $\frac{E_b}{N_0}$ is energy per bit to noise power spectral density ratio, and its value on the linear scale for 3×10^{-7} BER is taken as approximately 11 dB, assuming the modulation scheme used is BPSK [11]. The bit rate has to satisfy $R \leq C$, where C is the channel capacity and it is calculated as

$$C = B \log_2 \left(1 + \frac{S}{N}\right). \tag{1.3}$$

Finally, to establish a wireless link between the transceivers deployed on the racks u and v, $d_{uv} \leq 12$ has to be satisfied. Hence, to satisfy this in the proposed DC, as you can see in Fig. 1, there are 6 rows, and each contains 10 racks.

B. Wireless Link and Service Deployment (WSD) Problem Formulation

In this study, we embed virtual machines into physical machines considering the capacity of physical machines, and the source and traffic demand of virtual machines and establish dynamic wireless links between transceivers on the ToRs to meet the traffic demand. To embed virtual machines into physical machines, we introduce three capacity constraints:

$$\sum_{s \in S} \xi_{sp} \alpha_s \leq m_p \qquad \forall p \in P, \tag{2.1}$$

$$\sum_{s \in S} \xi_{sp} \beta_s \leq h_p \qquad \forall p \in P, \tag{2.2}$$

$$\sum_{s \in S} \xi_{sp} \gamma_s \leq c_p \qquad \forall p \in P, \tag{2.3}$$

where α_s, β_s, and γ_s are the memory usage, cache memory usage, and CPU usage of the virtual machine s, and m_p, h_p, and c_p are the total memory, maximum allowed cache size to be used, and maximum allowed CPU usage in percent of the physical machine p. The value of these parameters is known in advance. Furthermore, P is the global set of the physical machines, S is the global set of virtual machines run on these physical machines, and ξ_{sp} is an optimization decision variable indicating whether the virtual machine s is embedded into the

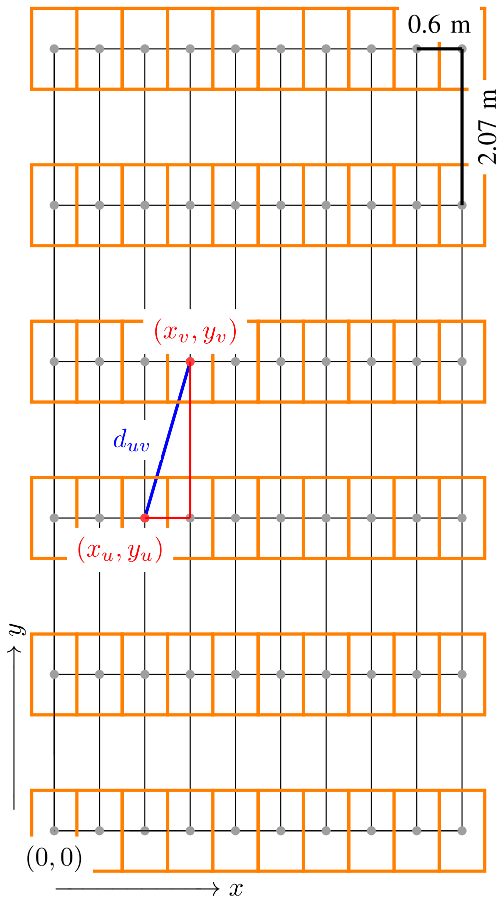

Fig. 1: Top view of the designed data center

physical machine p. If the virtual machine s is in the physical machine p, then ξ_{sp}=1, otherwise ξ_{sp}=0. A virtual machine can also only be embedded into one physical machine:

$$\sum_{p \in P} \xi_{sp} = 1 \qquad \forall s \in S. \tag{2.4}$$

We introduce a constraint to embed the virtual machines in a way that the total traffic demand between all virtual machines run in the physical machines deployed in any two racks cannot exceed the channel capacity of the wireless link established between these two racks, which helps to distribute the traffic demand of virtual machines evenly in the network:

$$\sum_{s,t \in S} \sum_{p,r \in P} \delta_{pu} \xi_{sp} X_{uv} \delta_{rv} \xi_{tr} r_{st} \leq C_{uv} \quad \forall u, v \in V, \tag{2.5}$$

where r_{st} is the average traffic demand in bps from the virtual machine s to t embedded in the physical machines p and r, respectively, and C_{uv} is the channel capacity of the wireless link established between the transceivers on top of the racks u and v. Furthermore, δ_{pu}, whose value is known in advance, indicates whether the rack u contains the physical machine p. If the rack u contains the physical machine p, then $\delta_{pu} = 1$, otherwise $\delta_{pu} = 0$. Moreover, X_{uv} is an optimization parameter indicating whether there is a wireless link between the racks u and v. If there is a wireless link between them, then $X_{uv} = 1$, otherwise $X_{uv} = 0$. There will be two-way communication between two transceivers, and a transceiver can have a wireless connection with only one transceiver:

$$\sum_{v \in V} X_{uv} \leq 1 \qquad \forall u \in V, \tag{2.6}$$

$$X_{uv} - X_{vu} = 0 \qquad \forall u, v \in V. \tag{2.7}$$

To establish a wireless link between two transceivers, $P_r \geq R_s$ has to be satisfied. We use this equality and the equations of (1.1) to (1.3) to present a relation between the received power and the capacity of the wireless link between two racks u and v as:

$$P_{rdB} \geq -187_{dB} + 10 \log_{10} (C_{uv}). \tag{2.8}$$

In the light of all constraints, the objective is to maximize throughput of the WDCN, deciding between which two racks there is a wireless link besides virtual machine embedding decisions:

$$\max_{X_{uv}, \xi_{pu}, \xi_{rv}} \sum_{s,t \in S} \sum_{p,r \in P} \sum_{u,v \in V} \delta_{pu} \xi_{sp} X_{uv} \delta_{rv} \xi_{tr} r_{st}.$$

IV. METHODOLOGY

Due to the complexity of our optimization problem, it takes considerable time to solve it with Mixed Integer Programming (MIP) solver. To decrease the time for solving the problem, we introduce two heuristic algorithms.

1) Greedy Heuristic - Heuristic for Wireless Link and Service Deployment (HWSD)
2) Local Search Heuristic - Improved HWSD (I-HWSD)

A. Heuristic for Wireless Link and Service Deployment (HWSD)

In HWSD, the aim is to choose the closest racks to establish a wireless link between and deploy services into physical machines in these racks according to their traffic demands. As described in Algorithm 1, HWSD takes the optimization model built before as an input. We first set the values of X_{uv} and ξ_{sp} as 0 for a fresh start. Then, we choose $rack1$ as the current rack and deploy the current service into the first physical machine of this rack, as seen in the lines 12, 17, and 18, respectively. A service is chosen as a current service if it has the biggest traffic demand to/from others. The rack closest to the current rack is chosen as the next rack and there will be a wireless link between them, seen in the lines 15 and 16. After deploying the current service, we choose the service deployed next as the service having the most traffic demand from/to the current one. If the link capacity between the current and the next rack is enough to deploy the service chosen and if we find a physical machine in the next rack having enough capacity for this service, we deploy it to the chosen physical machine, as shown between the lines 22 and 38. Every time we deploy a service to a physical machine, we need to update the heuristic value according to satisfied traffic demand and decrease the wireless link and physical machine capacities for further service deployment, as seen in the lines 29 and 31. To deploy the next service that has the most traffic demand to/from the current service to the next rack, we need to set the current rack as the next rack and vice versa. It helps services that have the most traffic between to be deployed to the racks having a wireless link. The deployment process continues until the link capacity between the current and the next rack or the capacity of physical machines deployed at these racks are not enough for further deployments. In this case, we need to choose new racks to establish a connection between and continue to the same process, as seen in the lines between 38 and 44. All processes end when all services are deployed. It is necessary to set start and current values of X_{uv} and ξ_{sp} whenever we establish a wireless link and deploy a service, respectively, to use them in I-HWSD.

B. Improved Heuristic for Wireless Link and Service Deployment (I-HWSD)

We implement a local search heuristic called Improved-HWSD (I-HWSD) to further improve the heuristic solution obtained by HWSD. I-HWSD takes as input the model whose decision variables X_{uv} and ξ_{sp} set by HWSD as seen in Algorithm 1, and sets these values as initial values. The aim is to start with wireless link establishment and service deployment scenarios obtained by HWSD and fix some decision variables while optimizing the others. First, we decide which rack will be fixed, and then we fix the value of the related decision variables of wireless links established from this rack by using the current value set before. We also fix some services' related decision variables using their deployment information obtained before. The number of fixed services is given as an input. In each round, we fix related decision variables of different racks and services. Each time we fix the values, we give the model with fixed and unfixed decision variables as an input to the Gurobi MIP solver and find an optimal deployment and wireless link establishment scenario. During the optimization, values of fixed variables cannot be changed while the solver finds optimal values of unfixed variables. If the objective value obtained is greater than the one found before, we update the heuristic value and set the values of X_{uv} and ξ_{sp} found as the initial values of the next round. This process continues for the given periods in proportion to the total number of services. The final heuristic solution is the one obtained after processes in HWSD and I-HWSD, respectively.

Algorithm 1 Greedy Heuristic - HWSD

Input: $model$
Output: heu_sol
X_{uv} ←: get_wireless_links($model$)
2: ξ_{sp} ←: get_vm_deployment($model$)
for $u, v \in racks$ **do**
4: X_{uv} ←: set_attr($Start$, 0)
X_{uv} ←: set_curr(0)
6: **end for**
for $s, p \in$ product($services$, $phy\_machines$) **do**
8: ξ_{sp} ←:set_attr($Start$, 0)
ξ_{sp} ←: set_curr(0)
10: **end for**
heu_sol ←: 0
12: $current\_rack$ ←: 1
$next\_rack$ ←: find_closest_rack($current\_rack$)
14: $current\_service$ ←: find_max_demanded_service()
X_{uv} ←: set_attr($Start$, 1)
16: X_{uv} ←: set_curr(1) , where $u, v = current\_rack, next\_rack$
ξ_{sp} ←: set_attr($Start$, 1)
18: ξ_{sp} ←: set_curr(1) , where $s, p = current\_service, P1$
decrease_pm_capacities($P1$, $current\_service$)
20: **while** ($current\_service$ is unvisited) **do**
$next\_service$ ←: find_next_service($current\_service$)
22: **if** (Link capacity between $current\_rack$ and $next\_rack$ is enough to deploy $next\_service$) **then**
pm_found ←: 0
24: **if** (There is at least one PM p in the $next\_rack$ r whose capacity is enough to deploy $next\_service$) **then**
pm_found ←: 1
26: ξ_{sp} ←: set_attr($Start$, 1)
ξ_{sp} ←: set_curr(1) , where $s = next\_service$
28: update(heu_sol)
decrease_link_capacities($current\_rack, next\_rack,$
30: $next\_service$)
decrease_pm_capacities(p, $next\_service$)
32: $current\_service$ ←: $next\_service$
exchange_values($current\_rack$, $next\_rack$)
34: **end if**
if pm_found **then**
36: **continue**
end if
38: **end if**
$current\_rack$ ←: choose_rack()
40: $next\_rack$ ←: find_closest_rack($current\_rack$)
X_{uv} ←: set_attr($Start, 1$)
42: X_{uv} ←: set_curr(1)
,where $u, v = current\_rack, next\_rack$
44: **end while**
return heu_sol =0

V. RESULTS AND DISCUSSION

In this section, we present the numerical solutions of our optimization model and heuristics. We solve the optimization model and do some numerical experiments using Gurobi Optimizer [15] version 9.1.1 build v9.1.1rc0 (mac64) on a device with 2 GHz Quad-Core Intel Core i5 with 16 GB RAM.

Due to the complexity of our MIP problem, solving it with Gurobi MIP solver takes considerable time, hence, we introduce HWSD and I-HWSD in Section IV. Here, we compare their performance with the performance of the Gurobi MIP solver in terms of the optimal solution and the time taken to find this solution. We also evaluate the impact of the distance between different racks and the traffic demand between virtual machines on the heuristic solution found by HWSD.

For the experiments:

- We have considered small-sized data centers generally used for educational purposes. Table II shows the small-sized data center designs used in our experiments. We experiment on the data centers with the different number of racks, servers, and services to be deployed and rack arrangement. We assume that each rack contains 5 servers. Rack arrangement shows the number of racks deployed in the column and row, respectively.

- We assume that physical machines (servers) deployed on the racks has 256 GB RAM and 20 MB cache memory.
- We generate the traffic demand between virtual machines, and their memory and CPU needs selecting uniformly from the value interval specified in Table I.
- We assume that there is a high capacity link between the physical machines deployed on the same rack and so don't take intra-rack communication into account.

We perform the numerical evaluation in two stages. First, we run HWSD on our optimization model, and then we run I-HWSD on the modified model by HWSD. We analyze their performance comparing with the Gurobi MIP solver.

1. HWSD: The aim of using HWSD is to decrease the time taken to solve the WSD problem by Gurobi MIP solver, so we run HWSD on the models built for each WDCN design showed in Table II. The throughput achieved by HWSD depends on the capacity of physical machines, traffic and source demand of virtual machines, and the distance between racks. We evaluate the impact of traffic demand change of the virtual machines on the throughput got by HWSD in Fig. 2. When we decrease the traffic demand by 20 and 30%, the throughput decreases. When we increase the traffic demand by 20 and 30%, the throughput increases, but the increase is less than 20 and 30%, respectively. This is because the channel capacity between two transceivers limits the traffic demand we can meet, which indirectly affects deployment decisions. This is also why throughput drops when the traffic demand of virtual machines increases when 5 racks are deployed. The design of racks in the DC also affects the throughput obtained by HWSD because the distance between each rack affects the channel capacity between two transceivers. Fig. 3 shows the impact of the increase in distance between two racks in the same row and the same column by 1 meter, respectively. An increase in distance between two racks decreases the channel capacity and throughput obtained by HWSD.

Gurobi MIP solver takes hours to find an optimal service deployment and link establishment scenario. With HWSD, we decrease the time tremendously and approach the maximum throughput by about 20% as shown in Fig. 4. Table III shows the elapsed time to find a heuristic solution using HWSD and I-HWSD in second and to find an optimal solution using the MIP solver. It doesn't take more than 1 second to find an optimal scenario using HWSD for each WDCN design. Hence, even if we consider using HWSD for mid and large-size data centers, it may take solving the problem a couple of minutes at maximum. Table III also shows how HWSD tremendously shortens the elapsed time to find an optimal solution by the MIP solver. Fig. 4 shows the throughput obtained by HWSD, I-HWSD, and the MIP solver. The blue line represents the HWSD, and for all WDCN designs used in the experiments, it is clear that it approaches the optimal solution found by the MIP solver by between 12 and 21%. If the concern is to get high throughput as much as possible in a short time, HWSD is the one that can achieve it without straying too far from the maximum throughput. It is also useful to meet dynamically changing traffic with its fast solution-finding mechanism.

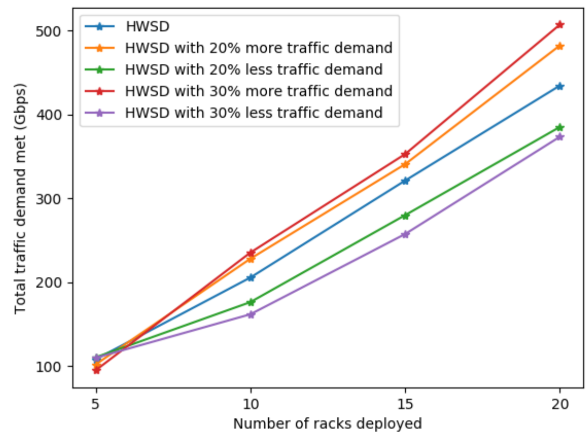

Fig. 2: The impact of traffic demand of the services on the throughput achieved by HWSD.

2. I-HWSD: It aims to increase the throughput obtained by HWSD and make it much closer to the maximum throughput found by the Gurobi MIP solver. We run I-HWSD on the model modified by HWSD for further improvements in the throughput. Fig. 4 shows how much I-HWSD improves the throughput found by HWSD and how much it brings the throughput closer to the optimal one found by the MIP solver. Unlike in HWSD, improving the solution takes quite a while seen in Table III. When the data center size increases, the time to obtain even small improvements is much more than in HWSD. The reason for taking much time of I-HWSD is that we use the Gurobi MIP solver to solve sub-problems constructed by I-HWSD, as explained in Section IV. Hence, when the model gets more complex, the elapsed time to improve the throughput increases markedly. Moreover, if we increase the running time of I-HWSD, we obtain better results. If the concern is to get higher throughput than the one found by HWSD without considering the time, I-HWSD can achieve it. The MIP solver still finds better throughput but spending much more time than HWSD and I-HWSD.

Our proposed heuristics provide an acceptable solution and

TABLE II: WDCN designs for the experiments.

# of Racks	Column x Row	# of Servers	# of Services
5	5x1	25	25
10	5x2	50	35
15	5x3	75	60
20	5x4	100	70

TABLE III: Time performance of HWSD, I-HWSD, and Gurobi MIP solver in second.

# of Racks	HWSD	I-HWSD	Gurobi MIP solver
5	0.01	32.5	1972.1
10	0.011	601.7	4982.3
15	0.072	1298.6	10193
20	0.085	1717.3	19589.9

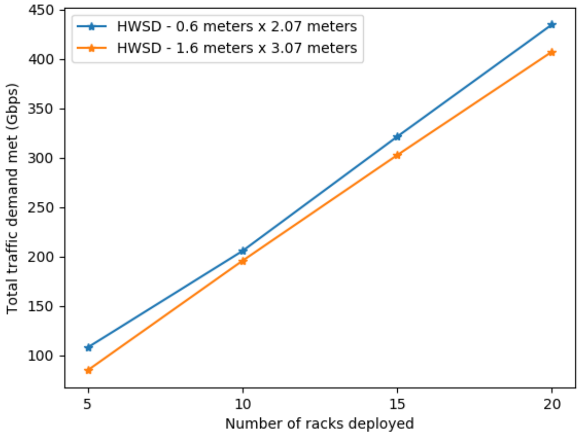

Fig. 3: The impact of distance between the racks deployed on the throughput achieved by HWSD.

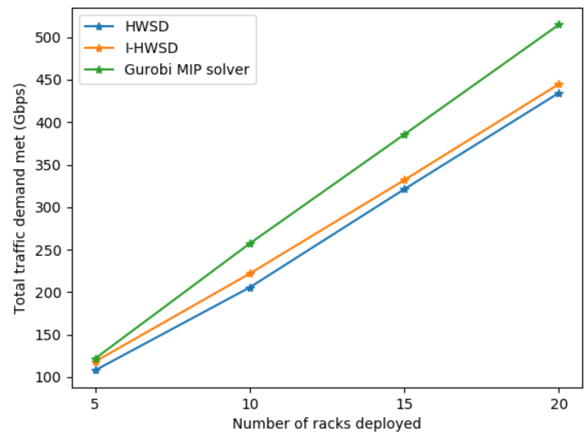

Fig. 4: The throughput achieved by HWSD, I-HWSD, and the Gurobi MIP solver.

a much more acceptable solution time compared to the Gurobi MIP solver. Besides the obtained throughput, our results show that with the WSD problem, we deploy the virtual machines into the physical machines distributing the traffic demands between different racks as much as possible evenly, and establish wireless communication between two racks that have more traffic demand for each other. This leads us to meet the notable inter-rack traffic demand in the WDCN.

VI. Conclusions and Future Work

In this work, we design a WDCN considering the wireless communication requirements to address the problems of wired DCNs. We propose an optimization model to solve our WSD problem that aims to maximize the throughput of the WDCN by efficiently deploying virtual machines into physical machines and by jointly establishing wireless links between two racks considering their contribution to the throughput. We propose two heuristics named HWSD and I-HWSD to decrease the elapsed time by the Gurobi MIP solver to solve WSD. Our results show that the proposed heuristics considerably improve the time taken by the Gurobi MIP solver approximating the maximum throughput by around 20%. Although we obtain significant results, we only work on small-sized DC models. Hence, to feel more confident in our solutions, as future work, we aim to work on mid-sized and large-sized DCs.

Acknowledgment

This study is partially supported by Vodafone Telecommunication Inc., Turkey within the framework of 5G and Beyond Joint Graduate Support Programme coordinated by Information and Communication Technologies Authority.

References

[1] S. Rommel, T. R. Raddo, U. Johannsen, C. Okonkwo, and I. T. Monroy, "Beyond 5G - wireless data center connectivity," in *Broadband Access Communication Technologies XIII*, B. B. Dingel, K. Tsukamoto, and S. Mikroulis, Eds., vol. 10945, International Society for Optics and Photonics. SPIE, 2019, pp. 125 – 133. [Online]. Available: https://doi.org/10.1117/12.2506269

[2] C. E. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, vol. C-34, no. 10, pp. 892–901, 1985.

[3] Y. Cui, H. Wang, X. Cheng, and B. Chen, "Wireless data center networking," *IEEE Wireless Communications*, vol. 18, no. 6, pp. 46–53, 2011.

[4] A. Celik, B. Shihada, and M.-S. Alouini, "Wireless data center networks: Advances, challenges, and opportunities," 2018.

[5] S. A. Mamun, S. G. Umamaheswaran, A. Ganguly, M. Kwon, and A. Kwasinski, "Performance evaluation of a power-efficient and robust 60 ghz wireless server-to-server datacenter network," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 1174–1185, 2018.

[6] M. Pedram, "Energy-efficient datacenters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 10, pp. 1465–1484, 2012.

[7] S. Mollahasani and E. Onur, "Evaluation of terahertz channel in data centers," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 727–730.

[8] E. Baccour, S. Foufou, R. Hamila, and M. Hamdi, "A survey of wireless data center networks," in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, 2015, pp. 1–6.

[9] C.-L. Cheng and A. Zajić, "Characterization of 300 ghz wireless channels for rack-to-rack communications in data centers," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 194–198.

[10] B. Cao, J. Zhao, P. Yang, Y. Gu, K. Muhammad, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Multiobjective 3-d topology optimization of next-generation wireless data center network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3597–3605, 2020.

[11] S. A. Mamun, S. G. Umamaheswaran, S. S. Chandrasekaran, M. S. Shamim, A. Ganguly, and M. Kwon, "An energy-efficient, wireless top-of-rack to top-of-rack datacenter network using 60ghz links," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2017, pp. 458–465.

[12] N. Hu and W. Zhou. How to choose an it rack. [Online]. Available: https://www.se.com/ww/en/download/document/SPD_VAVR-9G4MYQ_EN/

[13] S. K. Yong and C.-C. Chong, "An overview of multigigabit wireless through millimeter wave technology: Potentials and technical challenges," *EURASIP J. Wirel. Commun. Netw.*, vol. 2007, no. 1, p. 50, Jan. 2007. [Online]. Available: https://doi.org/10.1155/2007/78907

[14] R. Daniels and R. Heath, "60 ghz wireless communications: Emerging requirements and design recommendations," *Vehicular Technology Magazine, IEEE*, vol. 2, pp. 41 – 50, 10 2007.

[15] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com