

Efficient Beamforming for Mobile mmWave Networks

Sara Khosravi, Hossein S. Ghadikolaei, and Marina Petrova
 School of EECS, KTH Royal Institute of Technology, Stockholm, Sweden
 Emails: {sarakhos, hshokri, petrovam}@kth.se

Abstract—We design a lightweight beam-searching algorithm for mobile millimeter-wave systems. We construct and maintain a set of *path skeletons*, i.e., potential paths between a user and the serving base station to substantially expedite the beam-searching process. To exploit the spatial correlations of the channels, we propose an efficient algorithm that measures the similarity of the skeletons and re-executes the beam-searching procedure only when the old one becomes obsolete. We identify and optimize several tradeoffs between: i) the beam-searching overhead and the instantaneous rate of the users, and ii) the number of users and the update overhead of the path skeletons. Simulation results in an outdoor environment with real building map data show that the proposed method can significantly improve the performance of beam-searching in terms of latency, energy consumption and achievable throughput.

Index Terms—Millimeter-wave, beam alignment, spatial channel response, spatial correlation, beam-searching, mobile networks

I. INTRODUCTION

Millimeter-wave (mmWave) communication is regarded as a promising solution to support high data rate demands in the next generations of wireless networks [1]. To compensate for the severe propagation loss in this band, both transmitters (Tx) and receivers (Rx) rely on directional communications using a relatively large number of antennas, feasible thanks to the short wavelength. The use of directional communications, especially with analog or hybrid beamforming architectures [2], complicates the channel estimation and beamforming tasks, since the channel dimension is large and it is available through the eyes of the analog filters (which are low-rank and non-invertible). The problem becomes even more challenging in a mobile network, where the mobility demands frequent re-executions of the optimal beamforming task to overcome misalignment between Tx and Rx beams [3]. Therefore, there is a natural tradeoff between the total beamforming overhead (which is a function of the number of re-executions of the task) and the instantaneous rate of the user.

To address the problem in a stationary environment, the existing approaches usually search over a codebook (a set of potential beams) to find the optimal beam pairs for Tx and Rx. In particular, the existing mmWave standards [4], [5] define a multi-resolution codebook and use an exhaustive beam-searching algorithm to find the direction with the maximum link budget. However, this approach is time-consuming because it includes many iterations of sending pilot signals and waiting for control/feedback frames. Generally speaking, the overhead increases with the number of beam directions

and Tx/Rx may remain most of the time in the beam scanning phase rather than the data transmission phase [6]. Other approaches, such as sparsity-aware beamforming [7] or subspace estimation [8], face a similar problem: their overheads hinder their applicability in mobile mmWave networks. Although the recent compressive-sensing based approaches [7], [9], [10] need a logarithmic number of measurements in beam-searching, they do not work with the existing mmWave devices because they require an adopted phase-array antennas [10] or phase coherent measurements [9].

The spatial channel response (SCR) of mmWave systems has two fundamental properties. First, it is sparse in the angular domain [6], [7], [11], [12]. In other words, there are a few significant line of sight (LoS) and non-LoS (NLoS) paths between Tx and Rx, hereafter referred to as a *path skeleton* [6]. Due to the lack of SCR knowledge, Tx/Rx in the beam-searching phase scans all directions exhaustively to find the best beamforming/combining direction. A disadvantage of this method is that the overhead increases almost linearly with the number of beam directions. The second property of SCR is the strong correlation of the values in proximity locations as the Rx experiences almost the same scattering environment [6], [12], [13]. This is also known as the robustness of the channel second-order statistics to the small mobility [3].

Spatial correlation of mmWave SCRs have received a considerable attention in the recent years in order to alleviate the beamforming overhead. For example, Sur *et al.* in [6] presented a model that captures spatial and blockage-invariant correlation among beams to predict beam directions when human bodies block the links. However, this model is based on static links and it cannot capture the Rx mobility and is limited to an indoor environment.

Zhou *et al.* in [12] proposed a beam steering approach in an office environment that leverages the correlation of the mmWave spatial channel in near locations to predict and re-establish the blocked links. Although the results are promising in terms of throughput gain, the computational complexity of the proposed approach makes it impractical in real-world scenarios with cheap mmWave mobile devices. Moreover, the re-execution of the beam-searching is based on a constant Euclidean distance, which is very heuristic and may vary considerably in different user mobility and networking scenarios. Indeed, as we show in this paper, having a small Euclidean distance does not necessarily imply a small/negligible change in the SCR.

The existing approaches are either not suited to an outdoor

mobile network or are heuristic based, hindering a proper control of the tradeoff among the instantaneous achieved rate and beamforming update overheads in various user mobility scenarios.

In this paper, we address the problem of the beamforming overhead and investigate an efficient beamforming algorithm in an outdoor environment. Inspired by the path skeleton approach [6], we assume that every Tx maintains a database including path skeletons of different locations in its coverage areas. First, we consider a negligible overhead to construct and update this database and develop an algorithm to track the correlation between path skeletons in different locations through a specific trajectory. We show that our algorithm requires the minimum number of coarse beam-searching in comparison to the existing methods [5] while keeping the throughput and achieved rate in a near-optimal level. Moreover, we show that our approach is also efficient in terms of energy consumptions, computational and signaling complexity. We then characterize the cost of database maintenance and show that it is inversely proportional to the number of users. It suggests that as the number of users increases, the overhead for the database update decreases drastically and therefore we can optimize the aforementioned tradeoff at almost no cost.

The rest of this paper is organized as follows. We introduce the system model in Section II. We describe our proposed method to track the correlation between path skeletons in Section III. We also present the ray tracing simulation results of running our algorithm in a real urban environment in Section III. We consider the overhead of building and updating the database in Section IV. The summary of our findings is presented in Section V.

Notations: Bold upper-case \mathbf{X} , bold lower-case \mathbf{x} and normal font x denote matrices, vectors and scalars, respectively. For any vector \mathbf{x} (or matrix \mathbf{X}), $\|\mathbf{x}\|_2$, \mathbf{x}^\top and \mathbf{x}^H are its l_2 -norm, transpose, and conjugate transpose, respectively. For any integer L , we define set $[L] = \{1, 2, \dots, L\}$. \mathbf{I}_L denotes an $L \times L$ identity matrix.

II. SYSTEM MODEL

We consider the downlink of a mmWave network with multiple base stations (BSs) and user equipment (UEs). Without a loss of generality and to keep the notations simple, we focus on a specific UE and its serving BS. We consider BS as the Tx and UE as the Rx. We assume that the Tx location is constant but the Rx is mobile and moves through a specific trajectory. Blue and green lines in Fig. 1 show different Rx trajectories.

We assume that Tx and Rx utilize uniform linear arrays (ULA) with N_{Tx} and N_{Rx} antennas, respectively. The separation between both transmit and receive antenna elements is $\lambda/2$, where λ is the wavelength. We consider one radio-frequency chain both in the Tx and in the Rx. The channel follows a narrow band cluster model with L paths, and block fading, where the small scale fading is constant over a coherent interval (CI). The channel matrix $\mathbf{H} \in \mathbb{C}^{N_{\text{Rx}} \times N_{\text{Tx}}}$ in one CI

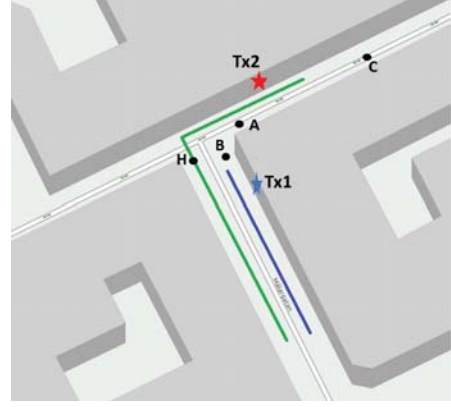


Fig. 1: Simulation area in the central part of Stockholm city. The blue star shows the location of the Tx1 and the red star shows the location of the Tx2. Blue and green lines illustrate the first and the second trajectory respectively. Point H shows the handover location.

can be expressed as [14]:

$$\mathbf{H}_{(x_i, y_i)} = \sqrt{\frac{N_{\text{Tx}} N_{\text{Rx}}}{L}} \sum_{\ell=1}^L \bar{g}_{\ell_i} e^{j2\pi f_{d_{\ell_i}}} \mathbf{a}_{\text{Rx}}(\theta_{\ell_i}) \mathbf{a}_{\text{Tx}}^\text{H}(\phi_{\ell_i}), \quad (1)$$

where $\bar{g}_{\ell_i} \sim \mathcal{CN}(0, 10^{-0.1PL_i})$ is the complex gain of the ℓ -th path that includes path loss and small-scale fading. PL_i is the omnidirectional path loss that is a function of the distance between the Tx and a specific Rx in location (x_i, y_i) where i is the location index. $f_{d_{\ell_i}}$ is the Doppler shift of the ℓ -th path that is characterized by the the direction of received paths relative to the motion of the Rx in location index i (see [7], [14] for more details). $\mathbf{a}_{\text{Tx}} \in \mathbb{C}^{N_{\text{Tx}}}$ and $\mathbf{a}_{\text{Rx}} \in \mathbb{C}^{N_{\text{Rx}}}$ are unit-norm array response vectors at the Tx and the Rx, respectively.

Since we use ULA, $\mathbf{a}_{\text{Tx}}(\phi_{\ell_i})$ is [2]

$$\mathbf{a}_{\text{Tx}}(\phi_{\ell_i}) = \frac{1}{\sqrt{N_{\text{Tx}}}} [1, e^{j\pi \sin(\phi_{\ell_i})}, \dots, e^{j\pi(N_{\text{Tx}}-1) \sin(\phi_{\ell_i})}]^\text{H}, \quad (2)$$

where $\phi_{\ell_i} \in [0, 2\pi)$ and $\theta_{\ell_i} \in [0, 2\pi)$ are angle of departure (AoD) and angle of arrival (AoA) of the ℓ -th path in the location index i , respectively. The array response vector at the Rx, $\mathbf{a}_{\text{Rx}}(\theta_{\ell_i})$, can be expressed in a similar fashion by replacing N_{Tx} and ϕ_{ℓ_i} by N_{Rx} and θ_{ℓ_i} , respectively. For the sake of notation simplicity, we drop subscript (x_i, y_i) from \mathbf{H} when it is clear from the context.

With the unit-norm beamforming vector $\mathbf{f} \in \mathbb{C}^{N_{\text{Tx}}}$ in the Tx and the unit-norm combining vector $\mathbf{w} \in \mathbb{C}^{N_{\text{Rx}}}$ in the Rx, the received signal of path ℓ in one CI is

$$y_\ell = \mathbf{w}_\ell^\text{H} \mathbf{H} \mathbf{f} s + \mathbf{w}_\ell^\text{H} \mathbf{n}_\ell, \quad (3)$$

where s is the transmitted symbol and $\mathbf{n}_\ell \sim \mathcal{CN}(0, \sigma^2)$ is the Gaussian noise vector. For the sake of simplicity, we consider s as the ℓ -th column of $\sqrt{P/L} \mathbf{I}_L$ where the total transmission power P is equally divided among L paths. Since \mathbf{w}_ℓ is unit-norm, we have that $\mathbf{w}_\ell^\text{H} \mathbf{n}_\ell \sim \mathcal{CN}(0, \sigma^2)$. We express the beamforming vector, \mathbf{f} , and the combining vector, \mathbf{w} , as

$$\mathbf{f}(\phi, N_{\text{Tx}}) = \frac{1}{\sqrt{N_{\text{Tx}}}} [1, e^{j\pi \sin(\phi)}, \dots, e^{j\pi(N_{\text{Tx}}-1) \sin(\phi)}]^\text{H}, \quad (4)$$

$$\mathbf{w}(\theta, N_{\text{Rx}}) = \frac{1}{\sqrt{N_{\text{Rx}}}} [1, e^{j\pi \sin(\theta)}, \dots, e^{j\pi(N_{\text{Rx}}-1) \sin(\theta)}]^\text{H}. \quad (5)$$

Equations (4) and (5) indicate that only the phase of elements in beamforming and combining vectors can be varied. We only consider horizontal (2D) beamforming/combining and the extension to 3D is straightforward. Each path can be LoS or NLoS.

We define a path skeleton as a set of L paths that approximate the SCR between the Tx and Rx. The path skeleton can be obtained after running exhaustive beam-searching in all directions and applying the so-called path skeleton construction procedure in [6]. The Tx and Rx then search over this skeleton, instead of an exhaustive search, to substantially reduce the cost of channel estimation and beamforming design. We assume that Tx (BS) can store the path skeletons in different locations of its coverage area in a database, which should be regularly maintained. However, this task involves two costs: the maintenance cost and the query cost. The maintenance cost is the overhead of building and updating the database. The query cost is the number of times we inform the Rx about a new path skeleton. We first focus on the case where the maintenance cost is negligible and optimize the Rx (user) rate experience with a given limited query budget. We then relax the negligible maintenance cost and propose a simple approach to update the database, whose cost is a decreasing function of the number of Rxs.

Formally, we model a path skeleton based on the large-scale parameters of each path $p_\ell = \{\theta_\ell, \phi_\ell, \tilde{g}_\ell\}$, $\ell \in [L]$ between a Tx in a fixed location and a mobile Rx in location (x_i, y_i) as

$$\text{PS}_{(x_i, y_i)} = \sqrt{\frac{N_{\text{Tx}} N_{\text{Rx}}}{L}} \sum_{\ell=1}^L \tilde{g}_\ell \mathbf{a}_{\text{Rx}}(\theta_\ell) \mathbf{a}_{\text{Tx}}^H(\phi_\ell) \quad (6)$$

where $\tilde{g}_\ell = \sqrt{PL_i}$. At this location, the large-scale parameters θ_ℓ , ϕ_ℓ and \tilde{g}_ℓ correspond to the AoA, AoD and the gain of the ℓ -th path.

In this work, we omit the interference effects of other Txs and multi-user scenarios and leave them as the future work.

III. PROPOSED ALGORITHM WITH NEGLIGIBLE DATABASE MAINTENANCE COST

In this section, we consider the low overhead PS database maintenance scenario. This assumption essentially means all the PSs are available a priori at every Tx for all locations inside its coverage areas. Consequently, we only focus on the query cost and develop optimization problems to maximize the Rx instantaneous rate given a fixed query budget. In summary, our solution approach measures the change in the PSs as the Rx moves and initiates a query automatically only when a substantial change in PSs, and consequently in the optimal beamforming vectors, is observed. We will consider the database maintenance cost in Section IV.

Notice that our proposed algorithm is based on the downlink transmission; however, it can be applied to the uplink case by replacing the roles of the beamformer (\mathbf{f}) and combiner (\mathbf{w}) and modifying the channel matrix \mathbf{H} .

A. Beamforming Design

In the pilot transmission phase, the Rx asks the database for the PS of its current location (x_i, y_i) . Then, a sequence of pilot

signals, $\text{Pilot}_{\ell_i} = (\frac{P}{L}, \mathbf{f}(\phi_{\ell_i}), \mathbf{w}(\theta_{\ell_i}))$, $\ell \in [L]$, are sent along L paths in the skeleton, whose cardinality is much smaller than the total number of paths between the Tx and Rx, to find the existing non-blocked paths. More specifically, Rx measures signal strength in all $p_\ell \in \text{PS}_{(x_i, y_i)}$ and constructs \mathbf{H} from (1). If all paths $p_\ell \in \text{PS}_{(x_i, y_i)}$ are blocked or weakened due to the presence of some potential obstacles or the dynamics of the environment, the path skeleton finder procedure is called to refresh this entry ($\text{PS}_{(x_i, y_i)}$) of the database.

In the data transmission phase, the beamforming vector \mathbf{f} and combining vector \mathbf{w} are designed to maximize the link budget, namely:

$$\underset{\mathbf{f}, \mathbf{w}}{\text{maximize}} \quad |\mathbf{w}^H \mathbf{H} \mathbf{f}|^2 \quad (7a)$$

$$\text{subject to} \quad \mathbf{f} \in \mathcal{F}, \quad (7b)$$

$$\mathbf{w} \in \mathcal{W}, \quad (7c)$$

where \mathcal{F} and \mathcal{W} are predefined beamforming and combining codebooks, respectively. Vectors \mathbf{f} and \mathbf{w} follow the functions given by (4) and (5). When there is no restriction on using any phase shift ϕ at the Tx and θ at the Rx and the number of antenna elements (N_{Tx} and N_{Rx}) grows large, the optimal solution of (7) is identical to the antenna response toward the strongest path. That is, $\mathbf{f}^* = \mathbf{a}_{\text{Tx}}(\phi_\ell)$ and $\mathbf{w}^* = \mathbf{a}_{\text{Rx}}(\theta_\ell)$, where $\ell = \text{argmax}_\ell \tilde{g}_\ell$. Notice that \tilde{g}_ℓ , $\ell \in [L]$ are found in the channel estimation over the PS. In summary, finding and applying the optimal beamforming in the asymptotic regime is very efficient: Rx should feed back to Tx the index of the path with maximum received signal strength. In this paper, we use this approach, which is asymptotically optimal in the sense of (7).

Given designed \mathbf{f}_ℓ and \mathbf{w}_ℓ , the received SNR follows

$$\text{SNR} = \frac{P |\mathbf{w}_\ell^H \mathbf{H} \mathbf{f}_\ell|^2}{BN_0}, \quad (8)$$

where, N_0 is the noise spectral density and B is the signal bandwidth in the data transmission phase. The achievable rate per second is then $\text{Rate} = B \log(1 + \text{SNR})$ and we can find the achievable throughput in location index i by multiplying Rate_i by the remaining data transmission time. A faster beam-searching leads to a longer data transmission time and perhaps a higher achievable throughput.

B. Tracking Spatial Correlation

To run the channel estimation (or equivalently beam-searching over the skeleton) on every new location, the Tx should inform the receiver about the set of paths in the skeleton. However, for most mobility models, the skeleton is almost the same over many CIs, essentially over several consecutive locations of the trajectory. To reduce unnecessary skeleton inquiry and feedback overhead, Tx continuously monitors the variations of the skeletons in different locations and reports the new one only when a significant change is detected. Formally, assume that Tx has already reported $\text{PS}_{(x_0, y_0)}$ as the reference skeleton. For any new location (x_i, y_i) , we define

$$d(x_i, y_i; x_0, y_0) = \|\text{PS}_{(x_i, y_i)} - \text{PS}_{(x_0, y_0)}\|_2. \quad (9)$$

Rx and Tx know $\text{PS}_{(x_0, y_0)}$ and $\mathbf{H}_{(x_0, y_0)}$. At any new location (x_i, y_i) , they use the skeleton of (x_0, y_0) to estimate $\text{PS}_{(x_i, y_i)}$ and $\mathbf{H}_{(x_i, y_i)}$. Then, Tx translates condition $d(x_i, y_i; x_0, y_0) \leq T$ to the validity of the existing skeleton. The beamforming vectors for (x_i, y_i) match to the strongest path in $\mathbf{H}_{(x_i, y_i)}$. If $d(x_i, y_i; x_0, y_0) > T$, Tx detects a significant change in the dominant paths of the channel, informs the Rx about the new skeleton, and asks for the beam-searching over the new skeleton. Then, this new skeleton is considered as the reference skeleton. This condition can be interpreted as if a set of pilot messages based on $\text{PS}_{(x_i, y_i)}$ are sent to the Rx in location index j . Since the distance between skeletons in location index i and j is higher than the threshold T , additional terms $\mathbf{a}_{\text{Tx}}^H(\phi_{\ell_j})\mathbf{a}_{\text{Tx}}^H(\phi_{\ell_i})$ appear in (6) and cause worse channel estimation result.

The performance of the proposed algorithm is dependent on the mobility model, network topology, and the decision threshold T . Among them we can only control T . Higher thresholds leads to fewer overheads of the feedback channel to send the new skeleton but also the adoption of suboptimal beamforming solutions: \mathbf{f} and \mathbf{w} are designed based on the skeleton in reference location ($\text{PS}_{(x_0, y_0)}$) not necessarily the skeleton at the current location (x_i, y_i) . Lower T improves the rate performance at the expense of higher feedback overhead. To properly set this hyper-parameter of the algorithm, we run the proposed algorithm on a dataset of various trajectories and find an optimal T from the following optimization problem:

$$T^* \in \underset{T > 0}{\operatorname{argmax}} \sum_{i \in [M]} \text{Rate}_i(T) \quad (10a)$$

$$\text{subject to } U < U_{\max}. \quad (10b)$$

where U_{\max} is a maximum tolerable number of skeleton queries (database query budget) and M is the length of the trajectory. The optimization problem (10) includes a one-dimensional search over $T > 0$ and can be solved numerically. Examples of the trajectories in the training dataset are shown in Fig. 1. Once T is obtained, this can be applied to users with similar mobility patterns.

Reference [12] considers $d(x_i, y_i; x_0, y_0) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$, Euclidean distance, to assess the PS correlation in different locations. This metric is both mobility-agnostic and topology-agnostic. To illustrate, consider the following example. Let Tx2 serves an Rx that is moving from location A to location B, see Fig. 1. Due to the small distance between location A and B (about 5 m), the Euclidean approach declares that the spatial channels in location A and B are highly correlated. However, the presence of a building obstacle between two locations completely changes the AoAs and AoDs and consequently the spatial channels. Thus, skeletons in location A and location B are not correlated. However, our proposed metric considers the actual distance between spatial channels (in norm-2 sense) and tracks the validity of the existing paths between the Tx and the Rx. Now, consider the Rx moves from location A to location C that is about 15 m far away. Although based on the Euclidean distance approach the spatial channel correlation weakens due to the long distance between the two locations,

Algorithm 1 Tracking spatial correlation

Inputs: A Trajectory = $\{(x_1, y_1), \dots, (x_M, y_M)\}$ of M coordinates, T and B .

```

1: Initialization: Set  $(x_0, y_0) \leftarrow (x_1, y_1)$ ,  $\text{PS}_0 \leftarrow$  skeleton at  $(x_1, y_1)$  and  $U = 0$ 
2: for  $i = 1, \dots, M$  do
3:   Send pilots over  $\text{PS}_{(x_0, y_0)}$  and observe  $\{\bar{g}_\ell\}_{\ell \in [L]}$ 
4:   if  $d(x_i, y_i; x_0, y_0) < T$  then
5:      $\mathbf{f}^* = \mathbf{a}_{\text{Tx}}(\phi_\ell)$ ,  $\ell = \operatorname{argmax}_\ell \bar{g}_\ell$ 
6:      $\mathbf{w}^* = \mathbf{a}_{\text{Rx}}(\theta_\ell)$ ,  $\ell = \operatorname{argmax}_\ell \bar{g}_\ell$ 
7:   else
8:     // Update reference point
9:      $(x_0, y_0) \leftarrow (x_i, y_i)$ 
10:     $\text{PS}_{(x_0, y_0)} \leftarrow \text{PS}_{(x_i, y_i)}$  and inform user
11:    Send pilots over  $\text{PS}_{(x_0, y_0)}$ 
12:     $\mathbf{f}^* = \mathbf{a}_{\text{Tx}}(\phi_\ell)$ ,  $\ell = \operatorname{argmax}_\ell \bar{g}_\ell$ 
13:     $\mathbf{w}^* = \mathbf{a}_{\text{Rx}}(\theta_\ell)$ ,  $\ell = \operatorname{argmax}_\ell \bar{g}_\ell$ 
14:     $U \leftarrow U + 1$ 
15:   end if
16:    $\text{Rate}_i = B \log(1 + \text{SNR}_i)$ 
17: end for
18: Outputs:  $\text{Rate}_i(T)$  and  $U$ 

```

the AoAs and AoDs in the two locations are quite similar so SCRs and equivalently PS should be indeed correlated. Hence, the norm-2 distance of channels as the metric to track spatial correlation in different locations can be much more accurate than a simple Euclidean distance approach.

C. Efficiency of the Proposed Algorithm

Efficiency can be defined based on four parameters: computational complexity, signaling complexity, throughput efficiency, and energy consumption. In the context of beamforming for mmWave networks, an efficient algorithm should keep the throughput and energy consumption at an optimal level with manageable computational and signaling complexities.

Given the presence of an updated skeleton database, Algorithm 1 adds negligible numbers of computations to the system. However, this computation uses historical data about the environment to decrease the number of running coarse beam-searchings in all directions and they trigger if the dynamics of the environment change the skeletons in the database. Moreover, based on our proposed beamforming design, in order to find the existing paths between Tx and Rx in reference locations, our algorithm searches only over the skeleton instead of all the space, leading to substantially less signaling complexity.

In terms of throughput, our algorithm checks the distance between spatial channels in different locations and updates the beamforming and combining vectors if they have been designed based on an obsolete channel, thereby guaranteeing a close-to-optimal performance of the channel estimation and beamforming design. Moreover, due to a faster beamforming design, we expect a longer data transmission time and therefore a gain in the achievable throughput.

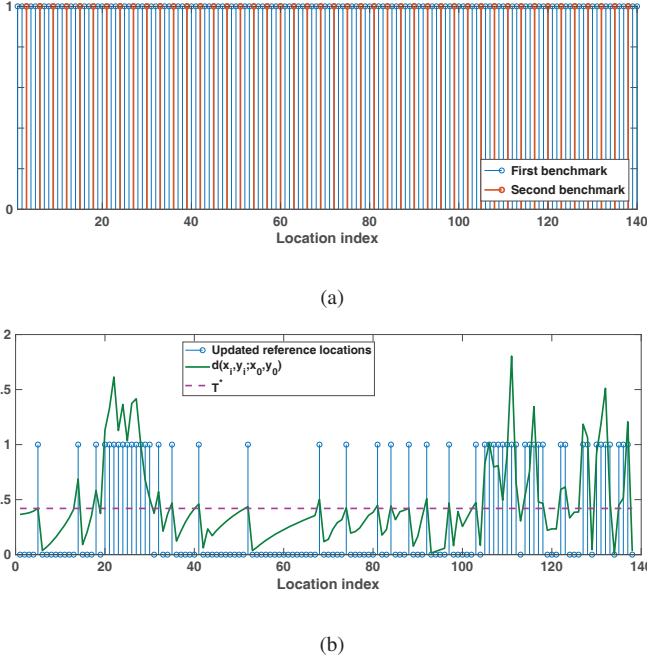


Fig. 2: The index of the updated reference locations by running (a) benchmarks, and (b) Algorithm 1.

In terms of energy efficiency, our approach keeps the number of pilots at a minimal level by searching only over the skeleton and reducing the database query frequency. Thus our approach is energy efficient, as we observe in the numerical results.

D. Numerical Results

We numerically evaluate the performance of our proposed approach in an urban environment using a ray tracing tool [15]. From the ray tracing output, we can obtain the existing paths between the Tx and the Rx in a specific location. To ensure high angular resolutions, we measure the AoAs and the AoDs with step sizes of 0.1 degrees. We extract the 200 m \times 200 m real building map of a central part of Stockholm city and use it as the input data for the ray tracing simulation. Fig. 1 shows the simulation area. We randomly assign glass or brick materials to the buildings. The general simulation parameters are listed in Table I. We consider two different Rx trajectories. We assume a normal pedestrian walk with speed 5 km/h.

1) *First Scenario*: In this scenario, we consider one Tx on the wall of a building with a height of 6 m and blue Rx

Parameter	Value
BS transmit power	30 dBm
Path loss exponent	3
Operating frequency	28 GHz
Signal bandwidth in data transmission phase	500 MHz
Thermal noise power	-174 dBm/Hz
Rx height	1.5 m
Tx height	6 m
Number of Tx antennas	8
Number of Rx antennas	8
Brick penetration loss [16]	28.3 dB
Glass penetration loss [16]	3.9 dB

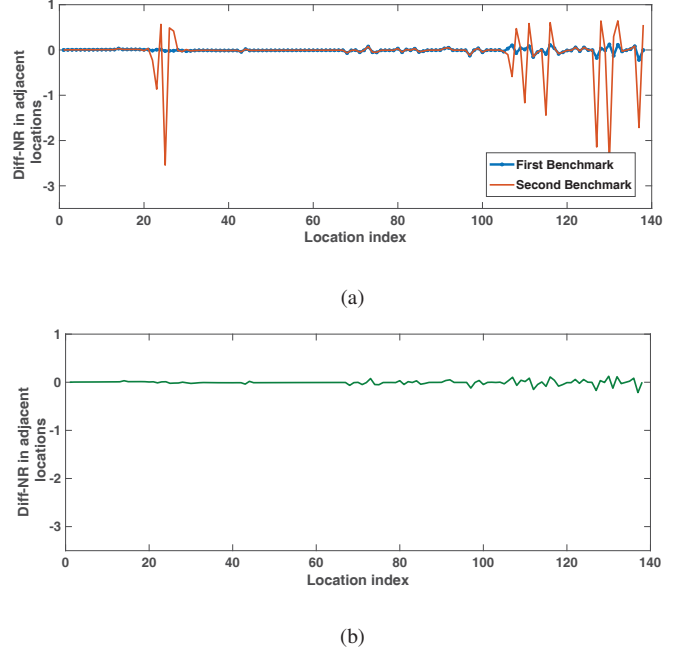


Fig. 3: Diff-NR in adjacent locations based on (a) benchmarks, and (b) Algorithm 1.

trajectory of Fig. 1. Rx is served by Tx1 along this trajectory. The trajectory length is 140 m, including 140 equidistant locations (once every meter), the Euclidean distance between Tx and Rx varies from 24 m to 119 m. In the *first benchmark*, we simulate a baseline where Tx extracts the PSs from the database in all locations so the database query budget is equal to the length of the trajectory. In the *second benchmark*, we simulate the approach of [12] where the beamforming and combining directions update based on a fixed Euclidean distance as shown in Fig. 2(a). The distance between two consecutive updates is 3 m, which is chosen to keep the same total number of updates as our approach (50 updates budget). In Fig. 2(b), the green curve shows distance between $PS_{(x_i, y_i)}$ in the location index i and the reference skeleton, $PS_{(x_0, y_0)}$. Based on Algorithm 1, the reference location is updated once the green curve is higher than the threshold T^* , set to be 0.42 for our environment as the optimal solution of (10) with a query budget of $U_{\max} = 50$. We have marked the update points by stems. The irregular inter-stem distance is due to the mobility of the user and similarity of the channels for some intervals.

We define difference normalized rate (Diff-NR) in two adjacent locations (x_i, y_i) and (x_{i+1}, y_{i+1}) as

$$\text{Diff-NR} = \frac{\text{Rate}_{i+1} - \text{Rate}_i}{\text{Rate}_{i+1}}. \quad (11)$$

In the first benchmark, clearly, Rx achieved rates in each location is maximum because of using optimal beamforming and combining directions in all locations but with the query cost of 140. In the second benchmark, As shown in Fig. 3(a), the user experiences some high fluctuations in the instantaneous rate, which are due to the fact that the channel correlation is sometimes weak before 3 m distance and we need to update

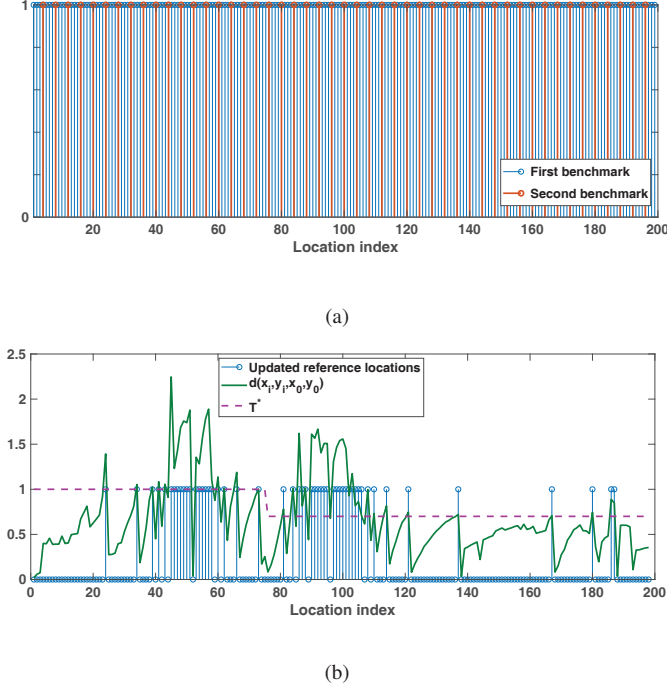


Fig. 4: The index of the updated reference locations by running (a) benchmarks, and (b) Algorithm 1.

the beamforming vectors in the meanwhile. Such fluctuations may prohibit the support of a reliable connection to the user. Fig. 3(b) shows Diff-NR of running Algorithm 1, indicating a very similar pattern as the first benchmark. In other words, Algorithm 1 with up to 50 queries to the skeleton database can perform as good as the first benchmark, where we have optimized the beamforming in all 140 locations of the trajectory.

2) *Second Scenario*: In this scenario, there are two TxS. Green line in Fig. 1 shows the Rx trajectory. The length of this trajectory is 200 m and contains two parts. In the first part Tx2 serves Rx. The second part is started from point H, where Tx1 starts serving the user after a handover. Again, we consider the baseline where each Tx queries PS sets from its database in all locations as the first benchmark and also consider the constant Euclidean distance query policy as a the second benchmark as shown in Fig. 4(a). The result of running our algorithm is presented in Fig. 4(b). In this case, we consider two different thresholds for each Tx. Based on numerical results, we choose $T_1^* = 0.7$ as the threshold of Tx1 and $T_2^* = 1$ as the threshold of Tx2 that are the solutions of the optimization problem in (10) with query budget 52. Reference locations in each Tx coverage area will be updated based on defined thresholds.

Fig. 5 illustrates Diff-NR in adjacent locations. It is evident that the rate fluctuation decreases after the handover point in location index of 75. In the first benchmark the query budget is 200 and the achieved rate in all locations is optimal. In the second benchmark the database query is requested every 4 m, shown in Fig. 4(a). Red curve in Fig. 5(a) shows the Diff-NR in adjacent locations that indicates multiple severe rate reductions as high as 10 times throughout the trajectory.

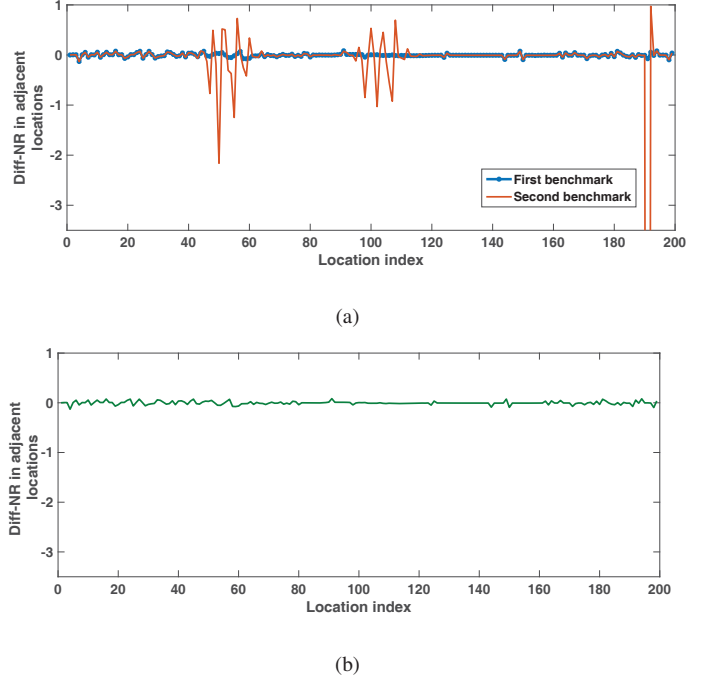


Fig. 5: Diff-NR in adjacent locations based on (a) benchmarks, and (b) Algorithm 1.

Indeed, Rx experiences low throughput in many locations of the trajectory, further highlighting that updating beamforming and combining vectors based on a constant Euclidean distance is not an efficient solution in a dynamic environment. Fig. 5(b) presents the results of running our proposed method, which implies that with running our algorithm and the query cost equal to 52, Rx achieved rate is near optimal in all locations of the trajectory.

IV. NON-NEGLIGIBLE DATABASE MAINTENANCE COST

In this section, we consider the database maintenance cost that includes the overhead of the building and updating database in a Tx. We start by describing database building and updating phases. Then, we show that the maintenance overhead is inversely proportional to the number of users. So in the outdoor environment with a large number of RxS, the overhead of building and updating the database could be almost negligible.

A. Database Construction

Initially, Tx divides its coverage area to smaller sections (grids) with pre-defined sizes as shown in Fig. 6. Tx assigns a unique ID to each grid, approximates each grid with one point, and considers one skeleton for that point. Therefore, the database has the same number of skeletons as the number of grid IDs. The grid size is equal in all sections and is chosen to balance the location resolution and the complexity of building a database. The beam-searching procedure runs once in each grid in order to build PS set of grid IDs.

The database has two lists: the normal list and the watch list. The normal list consists of updated PSs. The watch list

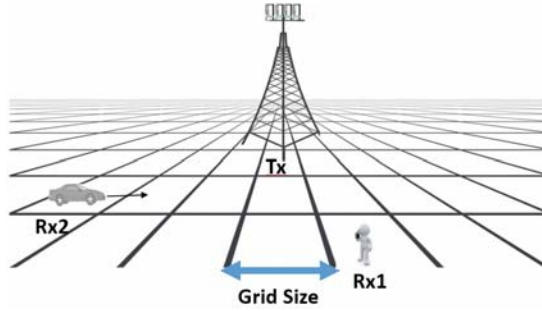


Fig. 6: Tx divides its coverage area to grids with an equal size.

includes the grid IDs whose PSs should be updated. Once a user is located in such grids, Tx may run the path skeleton finder procedure and update the PSs for those grid IDs. Clearly, all IDs are in the watch list initially. The main steps of the initial build up of the database are as follows:

- 1) Tx sends a skeleton finder request to all users in every grid ID, say x . If Tx distinguishes two or more than two Rxs in grid ID- x , it will choose one of them randomly.
- 2) The selected Rx confirms this request by sending a skeleton finder acknowledgment and the discover procedure (like the one in [6]) will be started afterward. Notice that an Rx may not confirm the request, for example, due to its low battery level.
- 3) The output of discovery process is the path skeleton in x , which includes all paths between Tx and the grid ID- x . Tx stores the skeleton, moves ID- x to its normal list, and activates an aging counter for this skeleton. The counters determine the age of each PS in the normal list.

Tx repeats the above process for all grid IDs of the watch list and stores them in the normal list of the database.

In the database updating phase, Tx checks the aging counters of the PSs in the normal list. If an aging counter exceeds a predefined threshold ($T\text{-Aging}$), Tx removes the ID from the normal list and adds it to the watch list of the database. It means that the skeleton discovery process runs again for this grid ID. Fig. 7 summarizes the proposed building and updating phases for a specific grid ID.

The $T\text{-Aging}$ depends on the environment. For a highly dynamic environment like crowded streets, it should be shorter than a stationary environment, as the skeletons may change more frequently in the former situation so Tx needs to update its database in a shorter periods.

B. Database Maintenance Overhead

In this subsection, we show that the overhead of building and updating the database is inversely proportional to the number of users in the environment. We then present numerical results of database maintenance overhead. In the database building and updating phases the number of the times that a Tx sends the skeleton finder request to a specific Rx in a trajectory is the main metric in analyzing the database overhead. We assume that Rxs always confirm the skeleton finder requests received from the Tx. In the dense urban environment, the number of Rxs is essentially high so the probability that one

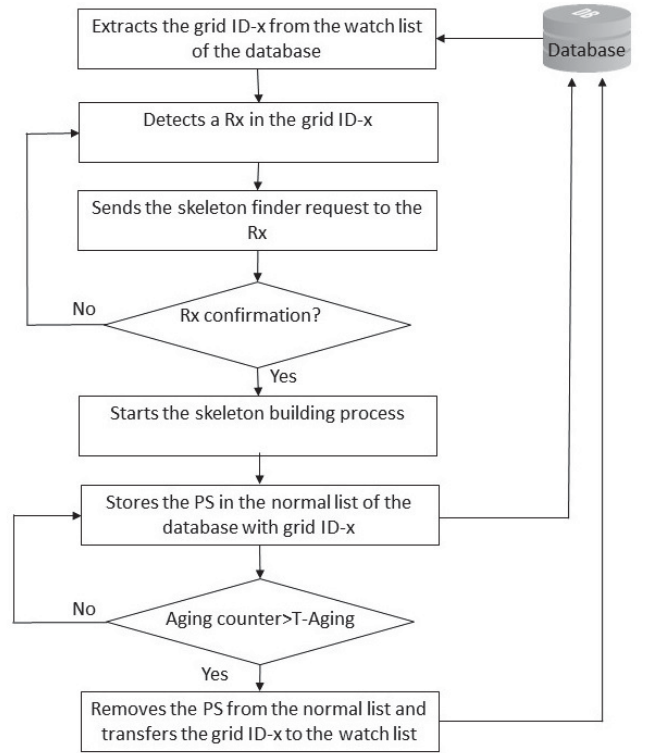


Fig. 7: A flowchart of the building and the updating database. The algorithm starts by extracting the grid IDs from the watch list of the database. When Tx detects a Rx in the grid ID, sends the skeleton finder request to it and stores the PS set in the normal list and activates the aging threshold. If the value of an aging counter exceeds the aging threshold, Tx transfers it (the grid ID) to the watch list again. In this case, the algorithm returns to the skeleton finder loop.

specific Rx receives several skeleton finder requests from the Tx can be low. In other words, database maintenance overhead is distributed among all Rxs in the Tx coverage area such that the database overhead decreases as the number of the Rx increase.

To have a better understanding of the interplay between the number of Rxs and the database maintenance overhead, we numerically evaluate the performance of the proposed method. We consider different sets of Rx in our studied environment in Fig. 1. We randomly assign different trajectories to each Rx. We assume each Rx is moving through its trajectory with some random speeds and directions. For instance in Fig. 6 we assign the normal pedestrian speed walk 5 km/h to Rx1 and average vehicle speed 30 km/h to the Rx2. Each Tx divides its coverage areas to grids with an equal size. We choose grid size equal to 2 m that is suitable for the crowded urban environment. We also consider the aging threshold equals to 2 minutes in our simulations. As mentioned in the previous subsection, grid size and aging threshold are highly dependent on the measurement environment and need to be very small in the dense urban environment.

We define C as the database overhead in a specific time duration. In other words, C is the number of the skeleton finder requests that the Tx sends to a specific Rx in its coverage area. First, we consider 100 Rxs that are moving

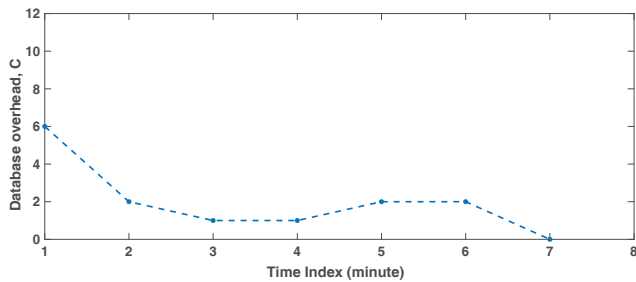


Fig. 8: The value C for a Rx that is moving in a trajectory in terms of simulation time index.

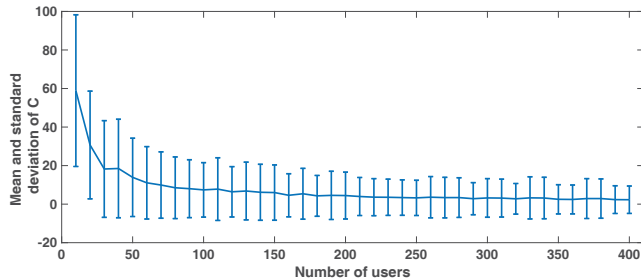


Fig. 9: Average and standard deviation of C for different number of Rxs/users.

through different trajectories with different speeds in our studied environment. All Rxs are located in different grids in Tx1's coverage area, see Fig. 1. First, we run the database building and updating phases. We consider Rx1, a pedestrian, that is moving through the trajectory that is shown with blue line in Fig. 1 and compute C during the simulation time, 7 minutes. Fig. 8 shows the values of C in different time indexes. In this case, the average amount of C is 2, which means Rx1 receives only 2 skeleton finder requests on average through its trajectory. In other words, the database overhead are divided between 100 Rxs in environment so mean values of C for each Rx is essentially low.

Now we sweep the size of Rx sets and repeat the previous simulation for each set. We plot the average amount of C for each Rx set in Fig. 9. We also plot the standard deviation of values C for each Rx set in Fig. 9 that expresses the amount of variation of different values of C for a specific Rx Set. From Fig. 9, it is evident that as the number of Rxs increase, the mean value of C and also standard deviation decreases, implying that the database overhead is almost negligible in crowded environments.

V. CONCLUSION

In this work, we proposed an efficient algorithm that leverages the correlation of path skeletons in order to decrease the number of running the coarse beam-searching methods in a mobile environment. We assume that each Tx is aware a priori of the path skeleton sets in its coverage area and show that the overhead of this assumption is almost negligible in dense urban environments. The simulation results highlight the efficiency of the proposed method and show that without significant reduction in the achieved rate, and almost the

same throughput, our method can decrease the number of running beam-searching. Moreover, the results confirm that re-execution beam-searching based on a constant distance may cause low throughput while our proposed method sustains the performance in the Rx trajectory.

REFERENCES

- [1] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez Jr, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, no. 1, pp. 335–349, May 2013.
- [2] S. Kuttu and D. Sen, "Beamforming for millimeter wave communications: An inclusive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 949–973, 2nd Quart, 2016.
- [3] H. Shokri-Ghadikolaei, C. Fischione, G. Fodor, P. Popovski, and M. Zorzi, "Millimeter wave cellular networks: A MAC layer perspective," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3437–3458, Oct. 2015.
- [4] "IEEE 802.15.3c Part 15.3: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs) amendment 2: Millimeter-wave based alternative physical layer extension," Oct. 2009.
- [5] "IEEE 802.11ad. Part 11: Wireless LAN medium access control MAC and physical layer PHY specifications - amendment 3: Enhancements for very high throughput in the 60 GHz band," Dec. 2012.
- [6] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "Beamspy: Enabling robust 60 GHz links under blockage," in *Proc. 13th USENIX Symposium on Networked Systems Design and Implementation (USENIX NSDI)*, 2016, pp. 193–206.
- [7] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, Apr. 2016.
- [8] H. Ghauch, T. Kim, M. Bengtsson, and M. Skoglund, "Subspace estimation and decomposition for large millimeter-wave MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 528–542, Apr. 2016.
- [9] Z. Marzi, D. Ramasamy, and U. Madhoo, "Compressive channel estimation and tracking for large arrays in mm-wave picocells," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 514–527, Apr. 2016.
- [10] H. Hassanieh, O. Abari, M. Rodriguez, M. Abdelghany, D. Katabi, and P. Indyk, "Fast millimeter wave beam alignment," in *Pro. the Conference of the ACM Special Interest Group on Data Communication (ACM SIGCOMM)*, 2018, pp. 432–445.
- [11] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir, "Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 4, pp. 1850–1859, Apr. 2013.
- [12] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *Pro. IEEE Conference on Computer Communications (INFOCOM)*, 2017, pp. 1–9.
- [13] S. Sun, T. S. Rappaport, M. Shafi, P. Tang, J. Zhang, and P. J. Smith, "Propagation models and performance evaluation for 5G millimeter-wave bands," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8422–8439, Sept. 2018.
- [14] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, Jun. 2014.
- [15] L. Simic, J. Riihijärvi, A. Venkatesh, and P. Mähoonen, "Demo abstract: An open source toolchain for planning and visualizing highly directional mm-wave cellular networks in the 5G era," in *Pro. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017, pp. 966–967.
- [16] H. Zhao, R. Mayzus, S. Sun, M. Samimi, J. K. Schulz, Y. Azar, K. Wang, G. N. Wong, F. Gutierrez, and T. S. Rappaport, "28 GHz millimeter wave cellular communication measurements for reflection and penetration loss in and around buildings in NewYork city," in *Pro. IEEE International Conference on Communications (ICC)*, 2013, pp. 5163–5167.