



BeamSpy: Enabling Robust 60 GHz Links Under Blockage

Sanjib Sur, Xinyu Zhang, and Parmesh Ramanathan, *University of Wisconsin—Madison*;
Ranveer Chandra, *Microsoft Research*

<https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/sur>

This paper is included in the Proceedings of the
13th USENIX Symposium on Networked Systems
Design and Implementation (NSDI '16).

March 16–18, 2016 • Santa Clara, CA, USA

ISBN 978-1-931971-29-4

Open access to the Proceedings of the
13th USENIX Symposium on
Networked Systems Design and
Implementation (NSDI '16)
is sponsored by USENIX.

BeamSpy: Enabling Robust 60 GHz Links Under Blockage

Sanjib Sur, Xinyu Zhang, Parmesh Ramanathan
University of Wisconsin-Madison

Ranveer Chandra
Microsoft Research

Abstract

Due to high directionality and small wavelengths, 60 GHz links are highly vulnerable to human blockage. To overcome blockage, 60 GHz radios can use a phased-array antenna to search for and switch to unblocked beam directions. However, these techniques are reactive, and only trigger after the blockage has occurred, and hence, they take time to recover the link. In this paper, we propose *BeamSpy*, that can instantaneously predict the quality of 60 GHz beams, even under blockage, without the costly beam searching. *BeamSpy* captures unique spatial and blockage-invariant correlation among beams through a novel prediction model, exploiting which we can immediately select the best alternative beam direction whenever the current beam's quality degrades. We apply *BeamSpy* to a run-time *fast beam adaptation* protocol, and a *blockage-risk assessment* scheme that can guide blockage-resilient link deployment. Our experiments on a reconfigurable 60 GHz platform demonstrate the effectiveness of *BeamSpy*'s prediction framework, and its usefulness in enabling robust 60 GHz links.

1. Introduction

The 60 GHz millimeter-wave (mmWave) band, with up to 7 GHz of unlicensed spectrum, offers the foundation for a new wave of applications, such as uncompressed video streaming, instant file sync, wireless data-centers and wireless fiber-to-home access. Demand for such data-hungry applications, together with worldwide availability of the 60 GHz band, have fueled multiple standardizations, e.g., IEEE 802.11ad [1], 802.15.3c [2] and ECMA-387 [3]. A similar paradigm has been advocated by industry to realize next-generation multi-Gbps cellular networks [4].

To counteract strong signal attenuation, a pair of 60 GHz transceivers can establish a *link* by forming highly directional *beams* using phased-array antennas. However, the pseudo-optical nature of a beam renders it extremely sensitive to blockage, especially in indoor deployments with heavy human activity [5]. Existing 60 GHz network standards have built in a counter-measure — a phased-array antenna can electronically switch between a prescribed set of beam directions, and bounce the signals off opportunistic reflectors, and thus detouring the blockage. But two new challenges emerge. (1) *Run-time overhead*. Searching for alternative Tx/Rx beam directions involves a tedious scanning and signaling proce-

sure. The overhead grows almost quadratically with the number of beam directions. Since a 60 GHz phased array typically generates tens to hundreds of beams, the overhead can easily overwhelm the precious channel time of a multi-Gbps link [6, 7]. (2) *Lack of outage prevention*. Beam searching/switching can only react after blockage occurs, which may have already caused detrimental effect on application and transport layer protocols.

In this paper, we propose *BeamSpy* to meet the above challenges. *BeamSpy* can predict the quality of alternative beams by only inspecting the channel response of the current beam used by the receiver. *BeamSpy*'s prediction mechanism leverages two fundamental properties of 60 GHz links: (1) *Channel sparsity*: no matter how many beam directions are available, the transmitter can only reach the receiver via a small set of dominating signal paths [7–10]. (2) *Spatial correlation*: the channels formed by different pairs of beams are often correlated, and the correlation remain unaffected by blockage, according to our measurement study (Sec. 3).

BeamSpy's prediction framework is model-driven. It exploits *channel sparsity* and models the channel between the transmitter and receiver using only a discrete set of signal paths, a set that we refer to as *path skeleton*. Furthermore, *BeamSpy* characterizes the *spatial correlation* by modeling the way different beam directions share the same *path skeleton*. The model takes into account the joint effect of phased-array beamforming and 60 GHz channel distortions. The modeling parameters are extracted from a one-time measurement, and invariant under blockage. Using *BeamSpy*, the Tx/Rx radios can instantaneously predict the best alternative beam direction whenever the current beams' quality drops.

BeamSpy can become a core engine for a wide range of 60 GHz protocols involving beam searching. In this paper, we apply *BeamSpy* to design two such protocols that facilitate reliable 60 GHz networking in blockage-prone indoor environment. (1) *Fast beam adaptation*: A link recovery protocol that can approximate the 802.11ad beam searching with a single implicit probing, thus evading the run-time overhead. (2) *Link outage prediction*. A risk-assessment algorithm that predicts the likelihood of link outage under blockage, thus offering guidelines for deploying 60 GHz links in a blockage-proof way.

We validate *BeamSpy* on a custom-built 60 GHz radio platform [7], along with trace-driven emulator running unmodified TCP/IP and application stack. Our ex-

perimental results demonstrate that: (1) *BeamSpy*'s prediction framework can effectively forecast quality of all beams by inspecting only one, with an average RSS prediction error of 0.02 to 1.2 dB. (2) *BeamSpy*'s fast beam adaptation protocol can efficiently identify the best beam pair, if any, that can overcome human blockage. It achieves comparable throughput performance with an oracle that knows exact beam quality, and outperforms 802.11ad significantly in application tests. (3) At deployment time, *BeamSpy*'s outage risk analysis can effectively assess vulnerability to human blockage and recommend a re-deployment when necessary.

General properties of 60 GHz channels, and the possibility of overcoming blockage via beam switching, are already well known [7, 11–13]. The key contribution of *BeamSpy* lies in a measurement-based and model-driven framework to help combat blockage without any extensive beam searching. In summary, our contribution breaks down into the following three aspects,

(1) **A measurement study to understand the unique properties of 60 GHz channel that are pertinent to the predictability of 60 GHz beamforming performance** (Sec. 3).

To the best of our knowledge, we are the first to perform a principled study of the *blockage-invariant spatial correlation* between 60 GHz beams, which roots in an interplay between extremely sparse 60 GHz spatial channels and phased-array beamforming.

(2) **A novel prediction framework that can capture the spatial correlation between beams using a *path skeleton* model, and can predict the performance of different beams without explicit probing** (Sec. 4). The framework is validated through extensive experiments on a 60 GHz testbed.

(3) **Applications of the prediction framework to design robust 60 GHz indoor networks** (Sec. 5). We design the first risk assessment protocol to predict the vulnerability of blockage at deployment time, and a fast beam adaptation protocol to efficiently overcome blockage at run-time.

2. Background & Motivation

2.1 60 GHz Channel and Impacts of Blockage

Due to smaller wavelength, a 60 GHz mmWave link suffers from 28 dB of higher propagation loss than a 2.4 GHz WiFi link. This disadvantage is compensated by using high-gain phased-array antenna, with multiple antenna elements that together act as a “focusing lens” to form highly directional RF beams. For instance, a 50×50 element phased-array can generate narrow beams of width 3° , providing an antenna gain of 36 dB [7, 14]. However, directional 60 GHz beams are highly susceptible to blockage [5, 7, 15] because of small wavelength, and hence limited ability to diffract around obstacles. When a beam is fully blocked by human body, link bud-

get is penalized by 20 to 30 dB [5, 7]. This is in stark contrast with directional links at lower frequencies [7]. Human movement in a room, therefore, can cause intermittent outage of 60 GHz links. Deployment in [5] observed 1% to 2% of link outage in an environment with 1 to 5 persons, and 14% to 22% with 11 to 15 persons.

The impact of link outage will be amplified at higher network layers. At the transport layer, TCP will respond by timing out, and may even need to re-establish connection [16]. Niche 60 GHz applications such as uncompressed video streaming may suffer from glitches and extended stalling period. Cable-replacement applications such as wireless HDMI, PCIe, USB deem outage event as cable-unplugging [17], and respond to it with very high re-establishment latency. As a result, it degrades the users' quality-of-experience significantly.

2.2 Limitations of Beam Scanning/Searching

A 60 GHz link may overcome blockage by switching Tx's and Rx's beams to form a detour path. Numerous beam-searching protocols, including the default in 802.11ad, has been proposed to search for the best beam pair [6, 18–20] However, any beam-searching protocol bears the following inherent limitations.

Overhead. Although dwelling on each beam direction only takes a few μs (including the time in signal, switching, and RSS feedback), the overhead increases almost quadratically with number of beams (48.4ms for a 16×16 phased-array antenna and 785.7ms for a 32×32 one [7]). In addition, beam-searching are usually invoked only at scheduled time-slots, and as such, only responds long after blockage already occurred.

Triggering threshold. Effective triggering of beam-searching remains an open problem. A typical approach is to invoke searching once link SNR changes beyond a threshold [12]. But under such an *aggressive mode*, search and switch may be triggered relentlessly throughout a blockage event, incurring substantial overhead [7]. Under a *conservative mode*, the device defers searching until link outage occurs. It then switches to best available beam pair, if any, to re-establish link. Albeit with less overhead, it may not be able to react quickly to blockage event which leaves the link at outage state for an extended period of time. It is virtually infeasible to set an optimal threshold which depends on elusive human blockage pattern.

Lack of preventive mechanism. Existing beam-searching protocols *react* to blockage, but do not afford any *preventive* mechanism to reduce the likelihood of link outage. An ideal preventive mechanism should be able to assess if a 60 GHz link is robust (*i.e.*, whether it can survive blockage using beam switching), and if not, suggest a re-deployment. However, such assessment is very challenging, because the effectiveness of beam switch-

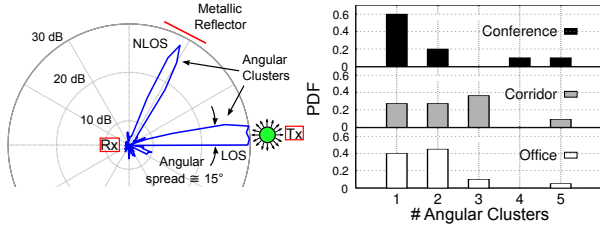


Figure 1: (a) Sparse channel response and strong clustering effect across different AOA. (b) Distribution of number of strong clusters for 50 links in 3 environments. ing highly depends on elusive environmental factors (reflectors positions, blocking positions *etc.* [7]). Interestingly, our experiments reveal that, maintaining proactive “backup” beams, commonly assumed in simulation models [11–13, 21], rarely helps in blockage prevention (Sec. 7). This is due to *spatial correlation*: multiple beams can be blocked simultaneously, even if they seem to point in different physical directions.

3. Measurements and Observations

In this section, we present four measurement observations of 60 GHz channels and phased-array beamforming. These observations constitute the foundation of *BeamSpy*’s prediction framework. Our measurement is conducted using a custom-built 60 GHz software radio platform (detailed in Sec. 6).

Observation 1 *Channel sparsity: 60 GHz channels are extremely sparse. The spatial channel response is dominated by a few paths from a few angular directions.*

The sparsity of 60 GHz channel is well known in prior measurement studies [7, 8, 15, 22, 23], and is presented here for completeness. Following a classical channel measurement approach [7, 8], we set up an omni-directional 60 GHz transmitter (Tx) in an office environment, while steering a 3° receiver (Rx) to resolve signal paths coming from different spatial angles. Fig. 1(a) plots an example *spatial channel profile*, *i.e.*, RSS measured along different Angle-Of-Arrival (AOA). The AOA pattern is extremely sparse — despite the omni-directionality of Tx, the Rx can only receive strong signals from a few densely concentrated directions (referred to as *angular clusters*), each spanning a narrow angle. Such sparsity is because *mmWave signal energy tends to concentrate around the LOS path and a few NLOS paths from strong reflectors*.

We have also measured 50 additional links randomly located in 3 different sites (office, corridor and conference room). Fig. 1(b) shows that all the links have 5 or fewer angular clusters, which again corroborates the sparsity.

Observation 2 *Spatial correlation: Given a 60 GHz phased-array with multiple beam directions, blockage of one beam affects the performance of other beams.*

For a phased-array receiver beam, the RSS along each spatial angle equals the corresponding beam gain pattern

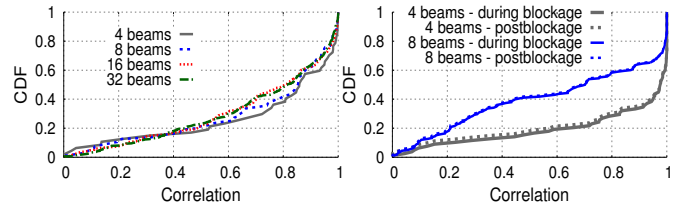


Figure 2: (a) Distribution of correlation between cluster blockage & RSS change of all beams. (b) Conditional distribution of correlation of RSS change of all other beams *w.r.t.* strongest beam during & after blockage.

multiplied by the channel gain. For example, under the same setup as in Fig. 1(a), Fig. 5(b) illustrates the spatial RSS when the receiver’s phased-array is tuned to 4 different beam patterns. Note that each beam pattern may contain multiple main lobes and sidelobes, but all 4 beams “share” the same two channel clusters. Therefore, when obstacles block a certain angular cluster, all beams may be affected in a correlated way.

To consolidate this intuition, we examine the impact of blockage on links using omni Tx and phased-array Rx, with 4 to 32 different beam patterns. For each beam, we measure the normalized RSS changes for different blockage positions and stack them into a vector. We then calculate the correlation coefficient between different beams’ vectors. Fig. 2(a) plots the CDF of pairwise correlation coefficients among 30 randomly deployed links. Around 80% of the links showed a strong inter-beam correlation (coefficient > 0.5), indicating that the *performance of many beams will change in a correlated way even when the blockage seems to land on one beam*.

Observation 3 *Blockage invariant correlation: The statistical correlation between different beams is invariant to human blockage.*

Intuitively, whether two beams suffer from correlated performance loss only depends on whether they strongly share certain angular clusters, and the correlation should not be affected by the blockage event itself. For verification, we repeat the previous experiment by blocking only the strongest angular cluster for each link. Meanwhile, we measure the RSS change of all beams. After removing the blockage, we repeat the measurement. Fig. 2(b) plots the distribution of the correlation between the performance change of all other beams *w.r.t.* the strongest beam. Evidently, *the spatial correlation between phased-array beams remain unaltered irrespective of blockage events*. *BeamSpy* essentially learns such correlation explicitly using a novel modeling framework (Sec. 4.2.1), and leverages the model to predict the best beam during blockage (Sec. 4.2.2).

Observation 4 *Human blockage does not create additional significant angular clusters.*

60 GHz signals are well known to be aquaphobic [7, 24, 25]. Therefore, when blocking a 60 GHz link, human

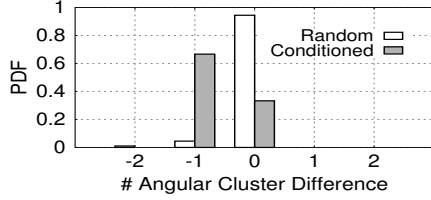


Figure 3: Distribution of difference in number of clusters before and during blockage.

body may absorb most of incoming signals [25], thus partially or completely blocking certain angular cluster(s).

This effect is evident in Fig. 3. Continuing with the prior setup, for each of the random blockages around each link, we identify the strong clusters and plot the PDF of the cluster number difference before and during blockage. A blockage may or may not fully block one cluster, evident from the result that the number of clusters can remain unchanged for more than 90% of the cases. However, if we condition on the cases where blockage landed on one or more clusters, about 63% links show 1 cluster less than those before blockage. Although signals can still bounce off the body, the end effect does not create a new strong/significant cluster.

4. BeamSpy Prediction Framework

Driven by the above observations, *BeamSpy* establishes a beam-quality prediction framework, designed for quasi-stationary 60 GHz links that may be occasionally displaced but frequently blocked by human movement [5]. Fig. 4 illustrates *BeamSpy*'s work flow. When a link is deployed, *BeamSpy* leverages full-beam scanning (such as in 802.11ad) to construct a novel *path skeleton* model, and extrapolate the blockage-invariant spatial correlation between different beams available on the Tx/Rx's phased-arrays. Afterwards, whenever beam quality changes due to blockage (indicated by SNR drop), *BeamSpy* can predict the quality of all other beams by simply measuring the Channel Impulse Response (CIR) of the beam in use.

In what follows, we first provide a primer on how 60 GHz signals are shaped by phased-array beamforming and channel response (Sec. 4.1), based on which we detail *BeamSpy*'s prediction framework (Sec. 4.2 and Sec. 4.3).

4.1 Joint Effect of Phased-array Beamforming and Angle Dependent Channel Distortion

A phased-array can apply a vector of beamforming weights to a set of omni-directional antenna elements to create directional beams, for either a transmitter or a receiver. For ease of exposition, we focus on a 1-D uniformly spaced antenna array, which has N antenna elements and can generate K beam directions/patterns in total. For the k^{th} beam, its *array-factor* (gain at spatial direction θ) is given by [24]:

$$A_k(\theta) = \sum_{n=1}^N \mathbf{w}(n,k) \cdot \exp(j2\pi n d \cos\theta / \lambda) \quad (1)$$

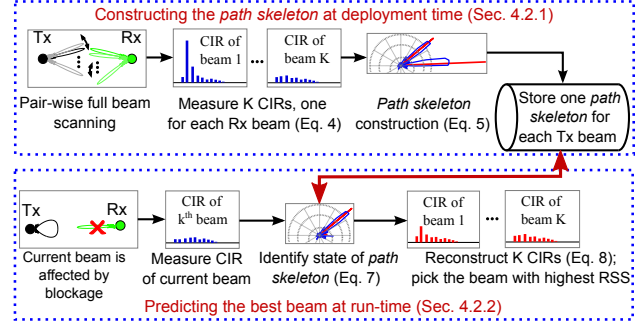


Figure 4: *BeamSpy*'s prediction framework.

where d is the antenna element spacing and λ the wavelength. $\mathbf{w}(n,k)$ is the beamforming weight applied to the n^{th} element when generating beam k . Each beam corresponds to one antenna gain pattern, with certain spatial directions amplified whereas others weakened.

For sake of simplicity, suppose the transmitter is omni-directional and receiver is a directional phased-array. When the receiver steers towards the k^{th} beam, its Channel Impulse Response (CIR), h_k , is a joint effect of the *array-factor* in Eq. (1) and the CIR of each antenna element. Suppose Θ is the maximum receiver aperture of each antenna elements (e.g., 360° for omni-directional ones) and P is the number of *paths* the signals can traverse between the transmitter and the receiver. Then, we have:

$$h_k = \sum_{\theta=0}^{\Theta} \sum_{p=1}^P A_k(\theta) \cdot \Gamma(p) \cdot \delta(\theta(p) - \theta) \quad (2)$$

where $\Gamma(p) = \beta(p)e^{j\phi(p)}$ is the channel distortion over path p , and $\beta(p)$, $\phi(p)$, $\theta(p)$ are the attenuation, phase and AOA of signal components traversing along path p . $\delta(\cdot)$ denotes the Dirac delta function capturing the effect of path directions on CIR h_k of k^{th} beam.

Intuitively, the CIR h_k captures the aggregated effect of all paths that arrive at each of the antenna elements, appropriately weighted by $\mathbf{w}(n,k)$ and summed together. In practice, not only the receiver, but also the transmitter's phased-array antenna will reshape the channel response, creating directionality effect along different spatial directions. This is equivalent to simply applying the transmitter's array-factor to each path in a reciprocal way.

When steered to beam k , a phased-array receiver can employ the built-in channel-training preambles such as in 802.11ad packets to extract the CIR value h_k , which is required in order to demodulate the packet payload [1].

4.2 Prediction Framework: Model & Algorithm

We formally define a *path skeleton* as the sparse set of dominating paths that can be used to approximate the spatial channel between a 60 GHz transmitter and receiver. Note that, the *path skeleton* only depends on the channel and is independent of the beamforming weights at Tx/Rx. However, the *overarching challenge* here is that the Tx/Rx can only measure the CIR when using a given beam, and cannot discriminate the channel distur-

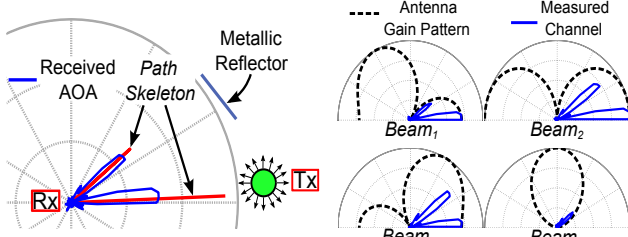


Figure 5: (a) Example path skeleton constructed by *BeamSpy*. (b) Measured CIR when receiver's phased-array antenna switches among 4 beam patterns.

tion/blockage along each specific path. But intuitively, since the channel is sparse, the very few number of dominating paths form a *path skeleton* that determines the performance of all receive beams. The core idea of *BeamSpy* is to “reverse-engineer” the *path skeleton* between the Tx and Rx (Sec. 4.2.1) and, when blockage occurs, estimate the blocked paths within the skeleton, and then predict the CIR of unobserved beams based on their known beamforming weights (Sec. 4.2.2).

4.2.1 Constructing the Path Skeleton

Fig. 4 (top) shows the one-time procedure to construct the *path skeleton*. When the 60 GHz AP and clients are deployed, *BeamSpy* invokes a full beam-searching procedure (such as in 802.11ad) once to capture the pre-blockage CIR between each pair of Tx and Rx beams. For each Tx beam, *BeamSpy* uses an array of K entries to store the CIR of the K receive beams.

Typical 60 GHz phased-array transmitters need to generate both highly directional and quasi-omni-directional beams [1, 7]. For simplicity, we only focus on quasi-omni transmit beam, whereas the receive beam can be any $k \in [1, K]$. *BeamSpy* constructs a *path skeleton* for each transmit beam. Simply put, the *path skeleton* consists of M paths arriving at receiver, which can be used to re-model the K CIR entries. These M paths should contain majority of the spatial information of the channel to represent the sparse set of signal clusters (*Observation 1*) between the Tx and the Rx.

So, how to construct the M paths, given that the receiver can only measure the CIR of each receive beam? *BeamSpy* solves this problem using a reverse-engineering model. Following Eq. (2), the CIR of beam k ($\in [1, K]$) can be represented as:

$$h_k^{rep} = \sum_{i=1}^M A_k(\theta_i) \cdot a_i e^{j\phi_i} \quad (3)$$

$$\text{and let } H^{rep} = \{h_1^{rep}, h_2^{rep}, \dots, h_K^{rep}\}$$

Here, each path p_i is represented by a triplet θ_i, a_i, ϕ_i , denoting its angle-of-arrival, amplitude and phase.

Denote the measured CIR of the K beams as:

$$H^{ms} = \{h_1^{ms}, h_2^{ms}, \dots, h_K^{ms}\} \quad (4)$$

Then, *BeamSpy* resolves the M paths as:

$$\{\hat{p}_1, \dots, \hat{p}_M\} = \underset{\{p_1, \dots, p_M\}}{\operatorname{argmin}} \|H^{ms} - H^{rep}\|^2 \quad (5)$$

Said differently, *BeamSpy* searches for M skeleton paths that can reproduce the measured CIR of all K beams with minimum error. This is a non-linear least mean square error curve fitting problem. We solve the problem using the *Levenberg-Marquardt Algorithm (LMA)* [26], which are widely applied in non-linear inverse problems.

When $M > K$, the problem (5) can become underdetermined, since we are fitting a skeleton of M paths to K measured samples. Therefore, given the receiver owns K beams, *BeamSpy* uses a skeleton of $M \leq K$ paths to represent the channel. In practice, $M \leq K$ holds because the number of strong angular clusters (paths) is typically below 5 in 60 GHz indoor/outdoor channels [9, 22] (c.f. Fig. 1(b)), whereas $K \geq 8$ even for a small 4-element phased-array [7]. We will further discuss about choice of M with system-level constraints in Sec. 4.2.2. Fig. 5(a) illustrates an example of *path skeleton* constructed using 4 angular CIR patterns, measured when the receiver is tuned to 4 different beam patterns (Fig. 5(b)).

4.2.2 Predicting the Best Beam

Fig. 4 (bottom) shows *BeamSpy*'s run-time prediction procedure. Specifically, given the measured CIR of current beam, *BeamSpy* estimates which paths within the *path skeleton* are affected by blockage. It then immediately predicts the quality of all alternative beams, based on the *a priori path skeleton* which captures the invariant spatial correlation between beams.

To identify the affected paths, *BeamSpy* makes the following approximation, inspired by *Observation 4*: human blockage annihilates existing skeleton paths, but does not create new paths. Suppose, the current receive beam index is k , with measured CIR h_k^{obs} . Let $b_i \in \{0, 1\}$ be an indicator variable denoting whether the skeleton path p_i is blocked. Following Eq. (3), we can represent the current beam's CIR as:

$$h_k^{pred} = \sum_{i=1}^M A_k(\theta_i) \cdot b_i \cdot a_i e^{j\phi_i} \quad (6)$$

where $b_i \cdot a_i$ models path p_i 's amplitude during blockage. Then we fit Eq. (6) to measured CIR h_k^{obs} to estimate $\hat{\mathbf{b}}$, the blocked/non-blocked states of all the skeleton paths:

$$\hat{\mathbf{b}} = \{\hat{b}_1, \dots, \hat{b}_M\} = \underset{\{b_1, \dots, b_M\}}{\operatorname{argmin}} \left\| h_k^{pred} - h_k^{obs} \right\|^2 \quad (7)$$

Finally, given the estimated $\hat{\mathbf{b}}$, and the pre-blockage *path skeleton* (Sec. 4.2.1), *BeamSpy* can reconstruct the CIR of any unobserved beam k' ($k' \neq k$):

$$h_{k'}^{pred} = \sum_{i=1}^M A_{k'}(\theta_i) \cdot \hat{b}_i \cdot a_i e^{j\phi_i} \quad (8)$$

The reconstructed CIR can be straightforwardly converted to link quality metric, like RSS or effective SNR [27], based on which *BeamSpy* can identify the best beam.

Note that, the *Minimum Mean Square Error (MMSE)* between current beam's measured and reconstructed CIR,

$$e_M = \min_{\{b_1, b_2, \dots, b_M\}} \left\| h_k^{pred} - h_k^{obs} \right\|^2 \quad (9)$$

characterizes the error when *BeamSpy* re-models blockage using $\hat{\mathbf{b}}$. This is a salient metric, later used as a *confidence level* of *BeamSpy*'s beam quality prediction.

Handling partial blockage and run-time execution. In practice, a path may consist of a cluster of angles, and human body may block only part of the cluster. To account for partial blockage, we quantize the elements $\hat{b}_i \in \hat{\mathbf{b}}$ into Q discretized levels and by default set $Q = 4$, corresponding to levels $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$. When multiple paths are affected concurrently (due to *either single or multi-person blockage*), the prediction framework is still applicable as multiple elements in $\hat{\mathbf{b}}$ may become 0 simultaneously. The non-linear problem in (7) may be solved using *LMA*, similarly to (5). However, the complexity may be too high for run-time execution. We simplify the solution using a *look-up table* approach. During the *path skeleton* construction phase, we build a table of size Q^M , each entry corresponding to the h_k^{pred} for a given configuration of $\hat{\mathbf{b}}$. At run time, to solve the problem (7), *BeamSpy* can simply look for the entry that matches h_k^{obs} with minimum error. To reduce the lookup time, we empirically limit M , the number of skeleton paths, to a maximum of 8. In practice, this empirical choice works because the typical number of angular clusters for 60 GHz indoor/outdoor channels is well below 5 [9, 22].

We emphasize that *BeamSpy* needs no PHY layer modification to the 60 GHz radios. It requires only channel response for each receive beam direction. Today's WiFi drivers already allow access to such information [27] and we expect this trend to continue for the 60 GHz drivers.

4.3 Operations of the Prediction Protocol

Lightweight prediction protocol. The above prediction framework focused on predicting the best Rx beam for a given quasi-omni Tx beam. We now describe how to extend the framework to predict the best *pair* of Tx and Rx beams. This is realized using a lightweight two-step signaling procedure illustrated in Fig. 6(a).

First, the AP temporarily switches to a quasi-omni antenna mode. The client measures the corresponding CIR using its current beam, and predicts the quality of alternative beams using the above algorithm. Then, the client feeds back the index of its new best beam direction (feedback sent via this new direction). Under the client's new beam direction, the AP predicts its new best beam, using the prediction algorithm in a reciprocal way.

The two-step signaling can be sneaked into existing MAC protocols, *e.g.* 802.11ad, as a background procedure. During normal data transmission, AP downgrades to quasi-omni mode, and invoke two-step signaling just like a normal Data-ACK exchange. *BeamSpy*'s prediction framework requires that the AP's quasi-omni beam cover its all possible fine beam directions, thus exciting all paths they can generate. Multiple quasi-omni beams

may be invoked separately to repeat the two-step signaling and satisfy this requirement. In practice, a quasi-omni beamwidth can be 60° to 180° , and 1 to 3 quasi-omni beams are sufficient to meet this goal.

Two additional points are worth discussion here:

- (1) *BeamSpy* requires that channel estimation is still feasible after blockage. This is reasonable because channel training preamble is much more robust than data payload. For example, 802.11ad uses a training sequence with a spreading factor of 128. Thus, channel estimation is still feasible even if link SNR is 21 dB ($10 \log_{10}(128)$) lower than the minimum SNR needed to demodulate data.
- (2) In the second signalling step, even if channel estimation is feasible, the client may fail to convey its best beam index to AP, *i.e.*, the signaling packet may be lost and AP times out. However, as long as the AP can extract the client \rightarrow AP CIR, it can recover the index as follows. It can search through each *path skeleton* corresponding to each of the client's Tx beams, and apply a *minimum Euclidean distance* metric to infer the client's best beam index j . Then, it applies *BeamSpy*'s prediction algorithm as if j is directly fed back by the client.

In practice, *BeamSpy*'s signaling failure rarely occurs, because it can employ a fast beam adaptation protocol (Sec. 5.1) to ensure the link migrates to the best alternative beam pair before the current one is fully blocked.

Refreshing path skeleton. After its initial construction, the *path skeleton* is refreshed on demand. *BeamSpy* can instantaneously detect staleness of the *path skeleton* using the *confidence level* parameter e_M (Eq. (9)). It re-runs the *path skeleton* construction (Sec. 4.2.1) if the e_M drifts from its initial value beyond a threshold (default to 20%). We note that the triplet θ_i, a_i, ϕ_i that characterize each path p_i are only affected by link distance, Tx/Rx orientation and strong reflectors in the environment. Human movement does not affect the parameters, because the effect is mostly weakening/blocking the paths, not creating new skeleton paths (*Observation 4*). Strong indoor reflectors (*e.g.* concrete walls, large metal furnitures *etc.*) are typically not changed frequently. So *BeamSpy* only needs to refresh the *path skeleton* at a very coarse time scale. We will evaluate the impact of environment change on the *path skeleton* update in Sec. 7.

5. Applications of BeamSpy

BeamSpy can potentially facilitate many higher layer protocols and network management schemes. A higher layer protocol may exploit *BeamSpy*-predicted RSS of beams to jointly adapt bit-rate and beam direction. A 60 GHz mesh network may use *BeamSpy* to instantaneously predict the best relay node that can help detour blockage. Further, network planners may use *BeamSpy* for what-if analysis when deploying 60 GHz links. In this section, we explore two salient applications of *BeamSpy* to en-

able robust and efficient 60 GHz networking. We use 802.11ad as a baseline MAC/PHY, but *BeamSpy* can be similarly integrated with other standards *e.g.* 802.15.3c.

5.1 Fast Beam Adaptation Under Blockage

We first employ *BeamSpy* to enable efficient beam switching under human movement and blockage. Specifically, *BeamSpy* acts as a meta-protocol to augment 802.11ad so as to quickly recover from link outage without the high-overhead beam searching [7].

Assuming the AP and client have established association, Fig. 6(b) illustrates the run-time state machine of the link. A legacy 802.11ad link transits between 3 states: *Norm*, *Outage*, and *Scan*. An 802.11ad link responds to the blockage event by triggering beam-scanning that searches for the best alternative beam pair. However, whether to use an aggressive or conservative threshold remains an open problem, due to a tradeoff between overhead and responsiveness (Sec. 2.2). *BeamSpy* overcomes the dilemma using its prediction framework together with the two-step signaling. Similar to aggressive mode, prediction is triggered when SNR changes. But, instead of a full scan, *BeamSpy* only needs to inspect its current beam quality, execute the aforementioned prediction algorithm, and estimate the best alternative Tx/Rx beam pair. Further, *BeamSpy* can help the link make an informed decision while utilizing its prediction result.

Recall that, *BeamSpy* outputs an *MMSE* (e_M) of model fit as *confidence level* of prediction (Sec. 4.2.2). Intuitively, a higher e_M indicates a larger error in estimating the blocked/non-blocked states of *path skeleton*, and thus poor prediction accuracy. We leverage e_M to make a probabilistic decision between utilizing *BeamSpy*'s prediction result and invoking an 802.11ad full-scanning. Specifically, we first use *BeamSpy* to predict the best available beam and evaluate e_M following Eq. (9). e_M is normalized *w.r.t.* to the current beam's channel gain $\|h_k^{obs}\|^2$, and the normalized value manifests how likely *BeamSpy* predicts wrongly. Then, we set the probability p of utilizing *BeamSpy*'s prediction result as $p = 1 - (e_M/\|h_k^{obs}\|^2)$. The probabilistic scheme reaps the benefit of accurate prediction, whenever possible, thus evading high overhead of beam-scanning. We will evaluate link performance gain from *BeamSpy*'s fast beam adaptation under intermittent human blockages in Sec. 7.1.2.

5.2 Link Outage Risk Assessment

BeamSpy does not guarantee a blocked link can be recovered via beam switching – such feasibility solely depends on Tx/Rx placement and environmental reflectivity [7]. However, *BeamSpy* can predict how likely a link deployment is to fail completely when blockage occurs. We call this scheme *risk assessment*. Risk assessment is critical when a 60 GHz link is deployed as a fixture, *e.g.*, from ceiling/wall to a furniture in home. Instead of

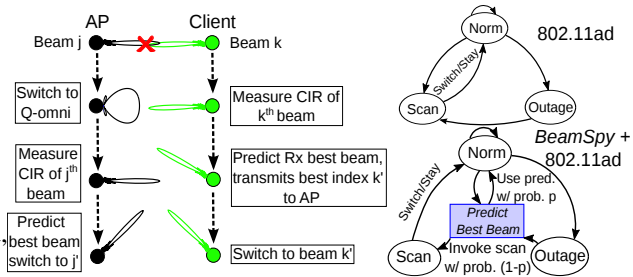


Figure 6: (a) Two-step signalling procedure (Sec. 4.3). (b) Radio state-transition of 802.11ad (top), and *BeamSpy*'s fast beam adaptation (bottom) that augments 802.11ad using the prediction framework.

a re-deployment (which may be unaffordable) upon link outage, at deployment time, the user can attempt different placement, orientation, or even place new reflectors, until the risk assessment mechanism indicates a low risk.

Risk assessment may be realized via empirical ways. For instance, one can conduct a war-driving and blocking the LOS to measure the likelihood of link outage. However, this can be tedious and cumbersome, since blocking different LOS positions may or may not fail a 60 GHz link with beam switching capability [7]. Alternatively, one can run an 802.11ad full scan and check if alternative Tx/Rx beam pairs can establish link connection besides the LOS beam pair. However, different beam pairs can suffer from *correlated outage* when blockage occurs (Sec. 3). This is again due to spatial correlation effect which can not be captured by empirical solutions.

With *BeamSpy*, we design a risk assessment mechanism to overcome such fundamental limitations. To capture the correlated blockage effect, we define the risk of a link deployment as a conditional probability,

$$\kappa = \mathbb{P}(\text{Link outage} | \text{Beam with highest RSS is blocked})$$

Link outage occurs if no beam sustains the RSS required by the lowest bit rate, even after beam switching.

To evaluate the κ -factor for a given deployment, *BeamSpy* first constructs the *path skeleton* and then “rehearses” all possible blockage patterns over the beam with highest RSS. Each blockage pattern is modeled by a quantized reduction of the amplitude of one or more paths that the highest-RSS beam traverses, and 4 quantization levels are used by default. Then, following Sec. 4.2.2, *BeamSpy* predicts quality of all beams under each possible blockage pattern, and computes deployment risk κ as fraction of cases where no beam can sustain the minimum bit rate. We summarize *BeamSpy*'s risk assessment procedure in Alg. 1. The algorithm is statistical in nature. It does not make any assumption about the human movement pattern or the absolute RSS drop due to blockage.

6. Testbed and Implementation

60 GHz Software-Radio Platform. To implement *Beam-*

Algorithm 1 Link Outage Risk Assessment

- 1: Initialize quantization level: $Q = 4$; $C_{Survival} = K \times Q^M$. $K =$ Number of beams, $M =$ Number of paths.
 - 2: **foreach** Tx beam
 - 3: Rx constructs *path skeleton* with M paths (Sec. 4.2.1);
 - 4: Initialize quantization matrix, C_M with Q^M entries;
 - 5: **foreach** row of $C_M \rightarrow q: [q_1, q_2, \dots, q_M]$
 - 6: Modulate *path skeleton* $[p_1, p_2, \dots, p_M]$ by quantization vector q ; Re-construct K beams from M modulated paths
 - 7: **If** RSS of all K beams $<$ *minimum RSS requirement*
 - 8: $C_{Survival} = C_{Survival} - 1$;
 - 9: Deployment risk, $\kappa = 1 - (C_{Survival}/K \cdot Q^M)$;
-

Spy and evaluate its efficacy, we use *WiMi* [7], a custom-built 60 GHz software radio platform. *WiMi* allows programmable waveform generation and received-signal processing on a PC host. Its RF front-end operates on 57-64 GHz carrier frequencies with 10 dBm output power and 245.76 Msps baseband sampling rate.

Due to lack of COTS 60 GHz phased-array antennas, we use a trace-driven approach to emulate channel response of a 60 GHz phased-array with a given set of beamforming weights. The foundation of this approach has been well established [6, 28]. Simply put, it conducts an angle-wise multiplication between the phased-array’s antenna gain, and the spatial channel response (*i.e.*, AOA pattern) between the transmitter and receiver. However, unlike the approach in [6, 28] that simulates the AOA patterns, we follow Rappaport *et al.* [8] to directly measure the AOA pattern of a 60 GHz link. Specifically, we equip *WiMi* with a highly directional 3° horn antenna, which is steered using a programmable real-time motion control system [29]. We measure the fine-grained AOA trace spanning 360° , and then convolve it with a standard 802.11ad beamforming codebook. This emulation approach is applied to each link in our experiment, with different blockage patterns, and different phased-array sizes that produce different number of beams. Note that, the mechanical movement of the horn antennas are used to resolve spatial channel and the movement itself does not affect the channel. Since spatial channel response is obtained by testbed measurement, a real phased-array may differ from emulated one only in its imperfect antenna gain pattern. As a validation, we measure transmitter beam pattern of the Wilocity 60 GHz radio [30] which has a built-in phased-array, and compare with the emulated one. The result (Fig. 7(a)) shows a close-match between these two. Also, we compare RSS of 15 different LOS links for both Wilocity radio (following approach in [31]) and *BeamSpy*-emulated. The *WiMi* transmitter and receiver were colocated with Wilocity for each link. Since, Wilocity radio does not allow us to control its beam direction, we only measure RSS of LOS strongest beam. Fig. 7(b) shows that *BeamSpy*-emulated RSS distribution follows the Wilocity distribution.

Due to hardware constraint, we can only use *WiMi*

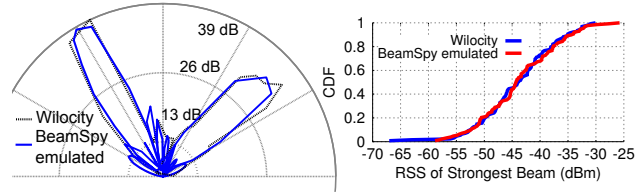


Figure 7: (a) Two example beam patterns of the Wilocity phased array [30], in comparison with the corresponding emulated beam patterns in our implementation. (b) Distribution of RSS of LOS strongest beam in Wilocity and *BeamSpy*-emulated links.

to send narrow-band (bandwidth=245.76 MHz) signals rather than 802.11ad-compatible one (1.7 GHz). The use of narrowband test equipment does not fundamentally affect our experimental validation, since directional mmWave channels experience minute frequency selectivity [22]. Even under a frequency selective channel, *BeamSpy* can be trivially extended by running prediction over different subchannels separately, and synthesizing the CIR to compute RSS across an entire band.

Finally, we implement *BeamSpy*’s prediction algorithm and applications within the software radios’ PC host, on top of an existing library that implements a virtual-clock-driven 802.11ad MAC [7]. The MAC module accurately follows 802.11ad’s default timing parameters when enforcing packetization with preamble, beaconing, beam-searching, inter-frame spacing, ACK, *etc.* The receiver measures RSS and noise floor on a per-packet basis, and translates it into achievable bit-rate following an 802.11ad specific rate table similar to the approaches in [7, 32].

Emulating Transport/Application Layers. We develop an emulation framework that can replay transport and application layer protocols, on top of fine-grained link-layer traces measured from *WiMi*. Our implementation adapts the popular *Dummysnet* emulator kernel [33]. During blockage, 60 GHz link quality (throughput and packet delay/loss) can vary significantly at fine time scales. However, *Dummysnet* can only configure link quality over coarse time scales through user space commands. Besides, packet losses can only be introduced probabilistically, which hinders accurate link-layer trace playback. To overcome such limitations, we augment the kernel to emulate link quality as functions of time at a fine granularity of 1 ms.

7. Evaluation

7.1 Micro-benchmarks

We now evaluate *BeamSpy*, focusing on three key performance questions: (1) How accurately can *BeamSpy* predict beam quality? (2) How much performance improvement can *BeamSpy* bring to a 60 GHz link under blockage? (3) How well can *BeamSpy* assess the risk factor of arbitrary 60 GHz link deployment?

7.1.1 Prediction Accuracy

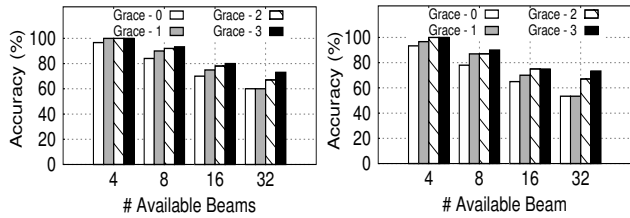


Figure 8: Accuracy of prediction. (a) Best Rx beam under a quasi-omni Tx beam. (b) Joint best transmit-receive beam pair. Grace- n indicates the oracle best beam lies within first $(n + 1)$ predicted best ones.

Accuracy of beam-quality prediction. We test link pairs deployed across 30 locations in an office. By default, a link runs a quasi-omni-directional transmitter. A human body statically blocks at a random position within LOS, and a 3° directional receiver captures AOA trace of the link before and during the blockage. Before blockage, the receiver emulates phased-array beams (Sec. 6) from the AOA trace and constructs the *path skeleton* (Sec. 4.2.1). During blockage, ground-truth RSS of all beams is measured first, but when running *BeamSpy*, the receiver only employs the CIR change of the single beam in use to predict the RSS of all other beams (Sec. 4.2.2).

Fig. 8(a) shows the accuracy of predicting the best receiver beam index for 4 phased-arrays with different sizes (*i.e.*, different number of antenna elements, and hence number of available beams). For a 4-beam receiver, *BeamSpy*'s mean prediction accuracy is well over 90%. The prediction accuracy drops to 71% as the number of available beams increases to 16, and 60% for 32. However, the oracle best beam still falls within the top 4 predicted beams with high probability (73% to 100%). We have also tested *BeamSpy*'s two-way signaling protocol, applicable when both the Tx and Rx use directional phased-arrays (Sec. 4.3). The results (Fig. 8(b)) show a similar level of accuracy as in predicting the best Rx beam.

The imperfectness in *BeamSpy*'s prediction stems from two factors: (1) The entire spatial channel response is represented by only few strong paths, which induces errors as the number of beams increase; (2) Approximating blockage impacts using quantized amplitude degradation. Despite this imperfectness, *BeamSpy* can substantially improve link robustness by acting as a meta-protocol for fast beam adaptation (Sec. 7.1.2), and as a risk assessment mechanism to guide link deployment (Sec. 7.1.3).

In practice, beam index matters less than quality of the beam *BeamSpy* predicts. Fig. 9(a) shows the absolute RSS difference between the predicted-best and oracle-best beam, which has a mean error of only 1.2 dB and max. 5 dB even for a 16-beam receiver. This implies that *even when BeamSpy predicts wrong best-beam index, the one it predicts does not have significant RSS difference from the oracle best*. Further, Fig. 9(b) plots the CDF of difference between the predicted RSS and measured

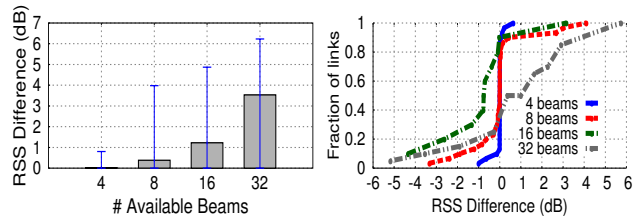


Figure 9: (a) RSS diff. between predicted and oracle best beam. (b) Predicted RSS diff. of the best beam.

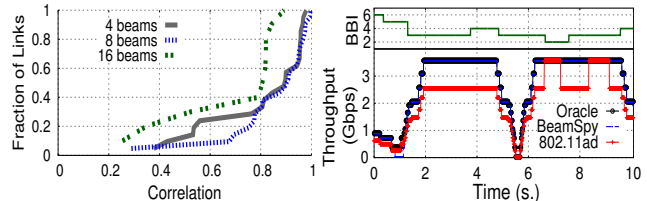


Figure 10: (a) Correlation between prediction accuracy and e_M metric. (b) Throughput for 10 s. random walking and blockages. Best Beam Index (BBI) for Rx is shown.

RSS, focusing on the best beam *BeamSpy* selects. We see that the 93-percentile error stays well within ± 4 dB, and median is below 0.8 dB even for the 16-beam case.

Confidence level of *BeamSpy*'s prediction. We evaluate whether the e_M metric in Eq. (9) can effectively indicate the prediction accuracy. We leverage the previous setup and for each blockage case, calculate *BeamSpy*'s prediction accuracy (in finding the best beam), while keeping track of e_M . Then, for each link, we find the correlation coefficient between the measured prediction accuracy and the e_M metric. Fig. 10(a) shows the CDF of correlation result across all links. We observe that for more than 90% of links (with 8 beams), the average correlation is greater than 0.5. Thus, *the e_M metric can indeed be used effectively as a confidence level indicator during BeamSpy's prediction*. A closer look to the rest 10% of the links indicates that their performance went below the lowest modulation level under blockage, and hence almost all beams perform similarly.

Effect of blockage position. We evaluate the prediction accuracy for different blockage positions for 20 link pairs. We setup the links with around 5m distance and block them at 10 approximately equal-spaced positions. Fig. 11(a) showcases the result as the position moves from near-Tx to near-Rx. A blockage close to either Tx or Rx blocks all the angular clusters which renders all beams' RSS to drop close to noise level. *BeamSpy* shows relatively lower prediction accuracy there, simply because all beams perform equally poorly. In all other cases, *BeamSpy* shows a high prediction accuracy consistent with our first micro-benchmark. We will evaluate the overall effect of random blockages on *BeamSpy*'s link-layer performance in Sec. 7.1.2.

Effect of quantization. Recall *BeamSpy* employs a quantized *look-up table* to model the effect of blockage (Sec. 4.2.2). Finer quantization helps capture nuances of real

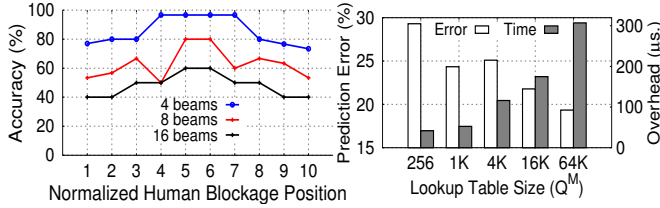


Figure 11: (a) Accuracy of prediction w.r.t. blockage positions. (b) Effect of increasing path skeleton lookup table size on prediction error and overhead.

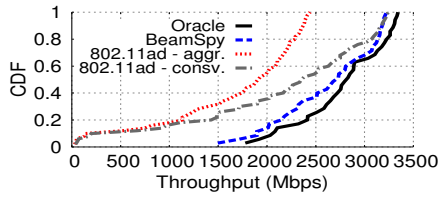


Figure 13: Throughput distribution of random walking and blockages in office environment.

blockage effect, but increases the look-up table size and computation overhead. Fig. 11(b) illustrates this trade-off using a 16-beam receiver as example. *BeamSpy*'s prediction framework runs on a desktop PC with 2.6 GHz CPU. With a small table, average prediction error is around 29% and prediction overhead is less than 40 μ s. As table size grows, the overhead grows proportionally, but the prediction error quickly drops below 20%. We believe a full-fledged optimized firmware/hardware implementation can help minimize *BeamSpy*'s overhead while maintaining its accuracy.

Temporal stability of path skeleton. We now evaluate *BeamSpy*'s path skeleton refreshment (Sec. 4.3) by generating a set of controlled events around the link. Fig. 12(a) shows an example of how the prediction accuracy varies over time when different events occur simultaneously with human blockages. Further, Fig. 12(b) quantifies the correlation between *BeamSpy*'s skeleton updates and the event types. Device displacement (mean angular displacement of 10 $^\circ$) or adding a new reflector usually triggers skeleton updates, since they may add new paths into the path skeleton. Small changes (involving books, kettle, laptops, chairs, etc.) hardly affect the sparse path clusters, and thus do not trigger an update of *BeamSpy*'s path skeleton. Overall, we can conclude that *BeamSpy* can maintain high prediction accuracy in quasi-stationary environment. It can adapt to infrequent displacement of the Tx/Rx or large reflectors, and is insensitive to small environmental changes.

7.1.2 Link Performance Gain from Prediction

In this section, we validate fast beam adaptation mechanism enabled by *BeamSpy* (Sec. 5.1). As benchmark comparison, we use 802.11ad with two triggering schemes. Under aggressive threshold, beam-scanning is invoked whenever current beam's SNR change leads to bit-rate

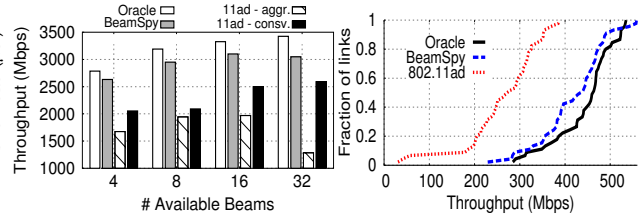


Figure 14: (a) Performance scaling with different number of available beams in office environment. (b) FTP throughput in office environment.

change; with conservative threshold, beam-scanning is only invoked when link cannot sustain the minimum bit-rate. We also compare with an oracle beam adaptation protocol that knows the best beam with no overhead.

Fig. 10(b) showcases how one link's throughput and best-beam index varies during random blockages. We see that *BeamSpy*'s throughput closely matches oracle. As a more general test, we repeat this experiment across 50 links deployed in the aforementioned 3 sites. Fig. 13 shows that *BeamSpy* provides a mean throughput gain of 957 Mbps and 479 Mbps (57% & 25%) over 802.11ad with aggressive and conservative thresholds, respectively. The correctness of the prediction model (Sec. 4.2) depends solely on channel sparsity and beam correlation, which are independent of locations (Sec. 3). Thus, the performance of *BeamSpy* remains consistently higher than 802.11ad, and close to oracle in different environments.

Fig. 14(a) shows how the link-level throughput scales with the number of available beams. *BeamSpy* performs close to the oracle with small to medium beam number, and 13.2% lower with beam number equal to 32. In all cases, *BeamSpy* outperforms the aggressive 802.11ad by 57–138% and conservative 802.11ad by 25–48%. Interestingly, due to high scanning overhead, the aggressive 802.11ad's performance drops substantially as beam number goes beyond 16, and much worse than the reactive/conservative approach that only responds to disconnections rather than SNR variations.

7.1.3 Performance of Outage Risk Assessment

We leverage the experimental setup in Sec. 7.1.1 to evaluate *BeamSpy*'s risk assessment Alg. 1. To assess the true blockage risk of a link deployment, we repetitively blocks the LOS of a link at random positions, and measure the fraction of cases where no beam can support the SNR needed for the minimum bit-rate. We compare *BeamSpy* with an 802.11ad-based risk assessment [12], which outputs "no risk" if it sees at least one additional backup beam that can establish the link before blockage.

Fig. 15 shows the results. Each dot in the scatter plot compares the *BeamSpy*-predicted risk (κ_p) with the ground-truth risk (κ_m). We observe that the backup-beam approach largely underestimates the risk. This effect is amplified in a high RSS regime where many beams can support good quality link before blockage. However, due

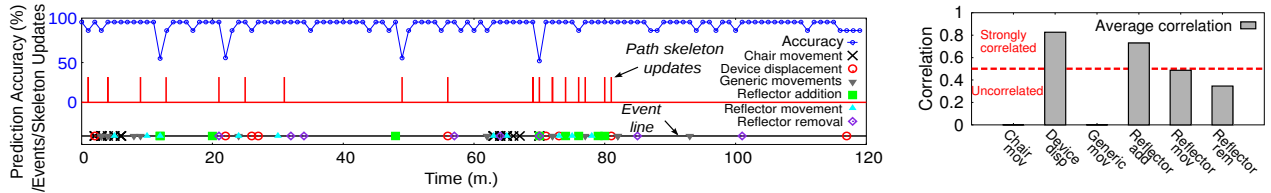


Figure 12: (a) Time-lapse of prediction accuracy and path skeleton updates under different events. (b) Average correlation between updates and event types. Generic movements include moving books, kettle and laptops.

to spatial correlation between beams (*i.e.*, many of them share an “invisible” set of skeleton paths), many of them can be blocked together. Taking into account the spatial correlation, *BeamSpy*’s risk prediction shows much closer match with the measured risk. Overall, *BeamSpy*’s predicted risk is slightly biased to the conservative side (on average 14% more risk than the measured one). However, a conservative assessment is more preferable than a overtly optimistic one, since it urges the deployment towards a blockage-proof stage.

7.2 Performance in User-level Applications

To test *BeamSpy* on real applications, we follow the prior setup to collect link rate traces for 50 random walking/blockage near a 8-beam link, and then load the traces into the emulator (Sec. 6). We compare *BeamSpy*’s fast beam adaptation with the conservative 802.11ad, which was shown to outperform the aggressive one.

FTP. We setup an FTP server using the emulator and client downloads a 650 MB file. For each of the downloads, we find mean throughput while a human walks by randomly, and repeat 10 trials. Fig. 14(b) shows resulting FTP throughput distribution, where *BeamSpy*’s median throughput is 147 Mbps (51%) higher than the 802.11ad-aggressive beam-searching.

Uncompressed Video Streaming. We set up a video server that transmits uncompressed video frames over the emulated network stack. The frames are captured and directly displayed through a VLC client. The supported video resolution varies from uncompressed standard definition (640×480) to Full-HD (1920×1080) at 25 fps. Fig. 16(a) shows that, under fixed Full-HD rate, 802.11ad suffers from high stalling durations during random human walking, and worst case stall can reach 460ms. In addition, under same median stall-duration, we measure mean video-rate that can be supported (Fig. 16(b)). *BeamSpy* provides $1.3\times$ improvement over 802.11ad. In summary, *BeamSpy* simultaneously boosts the video quality while reducing stallings, which can translate into quality-of-experience improvement for end users.

8. Discussion

Applying *BeamSpy* in outdoor environment. Our experiments have focused only on indoor 60 GHz links that are prone to human blockage. Correctness of *BeamSpy*’s prediction model depends solely on channel sparsity and blockage-invariant spatial correlation. The for-

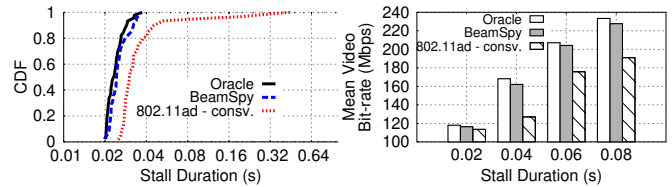


Figure 16: (a) Distribution of stall durations under fixed video bit-rate. (b) Mean achievable video bit-rate conditioned on same stalling rate.

mer has been extensively demonstrated in both short-range indoor and long-range outdoor environment [10, (Sec. III)]. The latter is a natural consequence of phased-array beamforming in sparse channels. Therefore, *BeamSpy*’s foundation still holds in outdoor environment. We leave the outdoor evaluation as future work.

Asymmetric antenna patterns. In certain 60 GHz deployment, the AP may use a larger phased-array antenna than client, thus creating asymmetric antenna patterns. We note that *BeamSpy* only requires each device to construct its own *path skeleton* to predict the alternate beam quality. Therefore, *BeamSpy* is still applicable under asymmetric phased-array antennas.

Handling client’s mobility. *BeamSpy* assumes quasi-stationary link deployment. If the client is mobile, the *path skeleton* construction has to be done at fine time scale, which is as costly as a full beam scanning in 802.11ad. However, by using a wide beamwidth at the mobile client and running *BeamSpy* at a static AP, *BeamSpy*’s *path skeleton* may still hold consistently. We leave the exploration of such mobile scenarios as future work.

***BeamSpy* in multi-links.** A strong interference from co-located links may affect *BeamSpy*’s *path skeleton* construction process and prediction results. However, when multiple devices are served by the same AP, the skeleton construction is already separated in time-domain owing to 802.11ad’s MAC protocol. Further, the prediction step is also mutually exclusive as an AP can serve only a single device at a time. Therefore, we expect *BeamSpy* can be easily extended to multi-links setup.

9. Related Work

60 GHz channel and network measurement. Using dedicated channel sounders, existing measurement studies have recognized the unique characteristics of the mmWave channel, especially the significant propagation loss [23] and vulnerability to human blockage indoor [5, 7, 15].

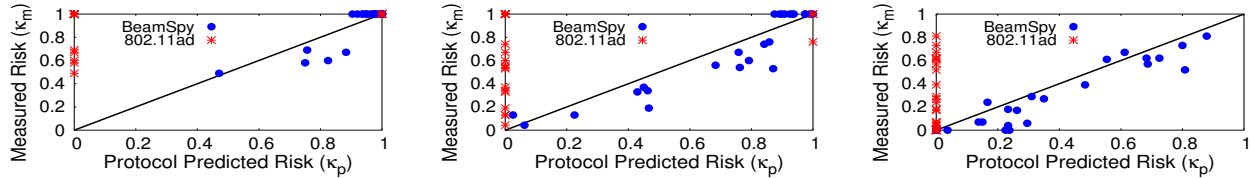


Figure 15: (a) Performance of BeamSpy’s risk assessment algorithm. (1) Low RSS regime (< -65 dBm), (2) Medium RSS (-65 to -60 dBm), (3) High RSS (> -60 dBm).

Consistent with our observations of channel sparsity, Xu *et al.* [8] and Sur *et al.* [7] showed that even if a transmitter is omni-directional, the received signals tend to be densely concentrated on a few angular clusters. For outdoor pico-cells, the blockage problem is less severe because of higher AP elevation, larger transmit power allowed by regulation and longer link distance [32] (and hence beamwidth expansion).

Efficient beam switching and channel tracking. Despite optimality, brute-force beam scanning incurs high overhead. More efficient beam scanning methods with approximately good performance have been proposed. They exploit hierarchical methods to reduce search space [6, 18, 19, 34]. 802.11ad and 802.15.3c both assimilated the idea, and patched refinement procedures to compensate for the sub-optimality of hierarchical search.

BeamSpy exploits blockage-invariant correlation between beams, a principle that has not been leveraged in previous works. In effect, *BeamSpy*’s fast beam adaptation can be integrated with these beam search protocols, in the same way as it does for 802.11ad (Sec. 5.1). Sparse channel distortion can be represented compactly using compressed sensing [35, 36]. A transmitter can leverage this principle to track the spatial channel response [37]. However, such compressive tracking still requires transmitter to scan multiple beam directions. In contrast, *BeamSpy* deals with link outage due to blockage, and can track the best among all beams by only examining current beam, without brute-force rescanning.

BeamSpy is partly inspired by CSpy [38], which exploited frequency domain correlation between adjacent WiFi channels to predict best channel without probing. Unlike *BeamSpy*, CSpy can only implicitly capture frequency domain correlation through extensive training of a machine learning model.

Surviving blockage. To overcome blockage, beam switching and reflection based methods have been proposed and validated by simulation [11, 12]. But the simulation ignored the practical correlation between beams/paths, which renders the “backup” beams or paths ineffective in practice (Sec. 7). Besides in-band beam searching, out-of-band solutions have been explored, *e.g.*, using a microwave channel [39]. Indirect out-of-band sensing methods like BBS [40] use 2.4 GHz MIMO to determine the spatial channel. Whereas it may help narrow down the search space, it still needs to steer across many 60

GHz beams to find the best-quality narrowest beam. This is because the number of antenna elements in a typical 60 GHz phased-array is much larger than a WiFi array, and thus there can be many narrow beams concentrated within one 60 GHz sector. In addition, like all existing beam searching methods, BBS is still a reactive mechanism and incapable of outage risk assessment. Alternatively, a detour path can be formed using relay nodes [14] which suffer less from reflection loss. But the dense deployment incurs additional cost.

Directional antenna networking at lower frequencies.

Directional-antenna networking has been extensively studied for the WiFi band, main focus being MAC design in ad-hoc networks [41–44]. More recent work advocates indoor high-speed networking using directional AP/clients [45,46]. Mechanically steerable antennas, and microwave band phased-array antennas have been used [47–49] to maintain directional connectivity with a mobile device. Blockage barely imposes any threat for low-frequency directional links, because of much wider beamwidth (45° or so), and lower penetration loss [7].

Acknowledgment

We thank Jialiang Zhang for help with *WiMi* hardware setup. The work reported in this paper was supported in part by the NSF under Grant CNS-1318292, CNS-1343363, CNS-1350039, CNS-1404613 and CNS-1518728.

10. Conclusion

We have provided experimental evidence of spatial correlation between beams generated by a 60 GHz phased-array antenna. This unique property originates from the fact that the beams can share a sparse set of propagating signal paths. We leverage this observation to design *BeamSpy*, a model-driven framework to predict the performance of multiple beams by inspecting the channel response of a single beam. *BeamSpy*’s modeling parameters are initialized through measurement, and are invariant under link dynamics caused by human blockage. We have validated the feasibility and effectiveness of *BeamSpy* on a 60 GHz testbed, and showcased how it enables efficient and robust 60 GHz networking under human blockage. We believe *BeamSpy* has wider implications for 60 GHz network design than what we have explored in this paper and can benefit a wide range of protocols involving phased-array beamforming.

References

- [1] IEEE Standards Association, "IEEE Standards 802.11ad-2012: Enhancements for Very High Throughput in the 60 GHz Band," 2012.
- [2] —, "IEEE Standards 802.15.3c-2009: Millimeter-wave-based Alternate Physical Layer Extension," 2009.
- [3] ECMA International, "Standard ECMA-387: High Rate 60 GHz PHY, MAC and PALs," 2010.
- [4] T. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. Wong, J. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" *IEEE Access*, vol. 1, 2013.
- [5] S. Collonge, G. Zaharia, and G. Zein, "Influence of the Human Activity on Wide-Band Characteristics of the 60 GHz Indoor Radio Channel," *IEEE Trans. on Wireless Comm.*, vol. 3, no. 6, 2004.
- [6] B. Li, Z. Zhou, W. Zou, X. Sun, and G. Du, "On the Efficient Beam-Forming Training for 60GHz Wireless Personal Area Networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, 2013.
- [7] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling," in *Proc. of ACM SIGMETRICS*, 2015.
- [8] H. Xu, V. Kukshya, and T. Rappaport, "Spatial and Temporal Characteristics of 60-GHz Indoor Channels," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 3, 2002.
- [9] P. Smulders, "Exploiting the 60 GHz Band for Local Wireless Multimedia Access: Prospects and Future Directions," *IEEE Communications Magazine*, vol. 40, no. 1, 2002.
- [10] T. S. Rappaport, E. Ben-Dor, J. N. Murdock, and Y. Qiao, "38 GHz and 60 GHz angle-dependent propagation for cellular and peer-to-peer wireless communications," in *IEEE ICC*, 2012.
- [11] Z. Genc, U. Rizvi, E. Onur, and I. Niemegeers, "Robust 60 GHz Indoor Connectivity: Is It Possible with Reflections?" in *IEEE Vehicular Technology Conference (VTC-Spring)*, 2010.
- [12] X. An, C.-S. Sum, R. Prasad, J. Wang, Z. Lan, J. Wang, R. Hekmat, H. Harada, and I. Niemegeers, "Beam Switching Support to Resolve Link-Blockage Problem in 60 GHz WPANs," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2009.
- [13] H. Zhang, C. Wu, X. Cui, T. A. Gulliver, and H. Zhang, "Low Complexity Codebook-Based Beam Switching for 60 GHz Anti-Blockage Communication," *Journal of Communications*, vol. 8, no. 7, 2013.
- [14] S. Singh, F. Ziliotto, U. Madhow, E. M. Belding, and M. Rodwell, "Blockage and Directivity in 60 GHz Wireless Personal Area Networks," *IEEE JSAC*, vol. 27, no. 8, 2009.
- [15] C. Anderson and T. Rappaport, "In-Building Wideband Partition Loss Measurements at 2.5 and 60 GHz," *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, 2004.
- [16] C.-Y. Huang and P. Ramanathan, "Network Layer Support for Gigabit TCP Flows in Wireless Mesh Networks," *IEEE Transactions on Mobile Computing*, 2014.
- [17] "Multi-gigabit, Low latency connectivity," <http://www.wi-fi.org/discover-wi-fi/wigig-certified>, 2016.
- [18] J. Wang, Z. Lan, C. woo Pyo, T. Baykas, C.-S. Sum, M. Rahman, J. Gao, R. Funada, F. Kojima, H. Harada, and S. Kato, "Beam Codebook Based Beamforming Protocol for Multi-Gbps Millimeter-Wave WPAN Systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, 2009.
- [19] Y. Tsang, A. Poon, and S. Addepalli, "Coding the Beams: Improving Beamforming Training in mmWave Communication System," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011.
- [20] K. Hosoya, N. Prasad, K. Ramachandran, N. Orihashi, S. Kishimoto, S. Rangarajan, and K. Maruhashi, "Multiple Sector ID Capture (MIDC): A Novel Beamforming Technique for 60-GHz Band Multi-Gbps WLAN/PAN Systems," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 1, 2015.
- [21] B. Gao, Z. Xiao, C. Zhang, L. Su, D. Jin, and L. Zeng, "Double-link beam tracking against human blockage and device mobility for 60-GHz WLAN," in *IEEE Wireless Communications and Networking Conference*, 2014.
- [22] T. Rappaport, F. Gutierrez, E. Ben-Dor, J. Murdock, Y. Qiao, and J. 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, 2013.
- [23] P. F. M. Smulders, "Statistical Characterization of 60-GHz Indoor Radio Channels," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 10, 2009.
- [24] T. S. Rappaport, R. W. H. Jr., R. C. Daniels, and J. N. Murdock, *Millimeter Wave Wireless Communications*. Prentice Hall, 2014.

- [25] S. Alekseev, A. Radzievsky, M. Logani, and M. Ziskin, "Millimeter Wave Dosimetry of Human Skin," in *Bioelectromagnetics*, 2008.
- [26] C.-T. Kim, J.-J. Lee, and H. Kim, "Variable Projection Method and Levenberg-Marquardt Algorithm for Neural Network Training," in *IEEE Industrial Electronics (IECON)*, 2006.
- [27] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," in *Proc. of ACM SIGCOMM*, 2010.
- [28] M. Park and P. Gopalakrishnan, "Analysis on Spatial Reuse and Interference in 60-GHz Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, 2009.
- [29] "Axis360 Motion Control System," <http://cinetics.com/two-axis360/>.
- [30] "Wilocity 802.11ad Multi-Gigabit Wireless Chipset," <http://wilocity.com>, 2013.
- [31] T. Nitsche, G. Bielsa, I. Tejado, A. Loch, and J. Widmer, "Boon and Bane of 60 GHz Networks: Practical Insights into Beamforming, Interference, and Frame Level Operation," in *Proc. of ACM CoNEXT*, 2015.
- [32] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao, and H. Zheng, "Demystifying 60GHz Outdoor Picocells," in *Proc. of ACM MobiCom*, 2014.
- [33] M. Carbone and L. Rizzo, "Dummysnet revisited," in *ACM SIGCOMM Computer Communication Review*, 2010.
- [34] K. Ramachandran, N. Prasad, K. Hosoya, K. Maruhashi, and S. Rangarajan, "Adaptive Beamforming for 60 GHz Radios: Challenges and Preliminary Solutions," in *ACM mmCom*, 2010.
- [35] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressed Channel Sensing: A New Approach to Estimating Sparse Multipath Channels," *Proceedings of the IEEE*, vol. 98, no. 6, 2010.
- [36] C. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of Compressive Sensing to Sparse Channel Estimation," *IEEE Communications Magazine*, vol. 48, no. 11, 2010.
- [37] D. Ramasamy, S. Venkateswaran, and U. Madhow, "Compressive Tracking With 1000-Element Arrays: A Framework for Multi-Gbps mm Wave Cellular Downlinks," in *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012.
- [38] S. Sen, B. Radunovic, J. Lee, and K.-H. Kim, "CSpy: Finding the Best Quality Channel Without Probing," in *Proc. of ACM MobiCom*, 2013.
- [39] H. Singh, J. Hsu, L. Verma, S. Lee, and C. Ngo, "Green Operation of Multi-Band Wireless LAN in 60 GHz and 2.4/5 GHz," in *IEEE Consumer Communications and Networking Conference (CCNC)*, 2011.
- [40] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with Eyes Closed: mm-Wave Beam Steering without In-Band Measurement," in *Proc. of IEEE INFOCOM*, 2015.
- [41] O. Bazan and M. Jaseemuddin, "A Survey On MAC Protocols for Wireless Adhoc Networks with Beamforming Antennas," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, 2012.
- [42] R. R. Choudhury and N. H. Vaidya, "Deafness: A MAC Problem in Ad-Hoc Networks when using Directional Antennas," in *IEEE ICNP*, 2004.
- [43] R. R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," in *Proc. of ACM MobiCom*, 2002.
- [44] R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, "On Designing MAC Protocols for Wireless Networks Using Directional Antennas," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, 2006.
- [45] M. Takai, J. Martin, R. Bagrodia, and A. Ren, "Directional Virtual Carrier Sensing for Directional Antennas in Mobile Ad Hoc Networks," in *Prof. of ACM MobiHoc*, 2002.
- [46] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "DIRC: Increasing Indoor Wireless Capacity Using Directional Antennas," in *Proc. of ACM SIGCOMM*, 2009.
- [47] A. Amiri Sani, L. Zhong, and A. Sabharwal, "Directional Antenna Diversity for Mobile Devices: Characterizations and Solutions," in *Proc. of ACM MobiCom*, 2010.
- [48] C.-F. Shih and R. Sivakumar, "FastBeam: Practical Fast Beamforming for Indoor Environments," in *International Conference on Computing, Networking and Communications (ICNC)*, 2014.
- [49] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das, "MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access," in *Proc. of ACM MobiSys*, 2007.