



Many-to-Many Beam Alignment in Millimeter Wave Networks

Suraj Jog, Jiaming Wang, Junfeng Guan, Thomas Moon,
Haitham Hassanieh, and Romit Roy Choudhury, *UIUC*

<https://www.usenix.org/conference/nsdi19/presentation/jog>

This paper is included in the Proceedings of the
16th USENIX Symposium on Networked Systems
Design and Implementation (NSDI '19).

February 26–28, 2019 • Boston, MA, USA

ISBN 978-1-931971-49-2

Open access to the Proceedings of the
16th USENIX Symposium on Networked Systems
Design and Implementation (NSDI '19)
is sponsored by



Many-to-Many Beam Alignment in Millimeter Wave Networks

Suraj Jog, Jiaming Wang, Junfeng Guan, Thomas Moon, Haitham Hassanieh, Romit Roy Choudhury
University of Illinois at Urbana Champaign
{sjog2, jw27, jguan8, tmoon, haitham, croy}@illinois.edu

Abstract

Millimeter Wave (mmWave) networks can deliver multi-Gbps wireless links that use extremely narrow directional beams. This provides us with a new opportunity to exploit spatial reuse in order to scale network throughput. Exploiting such spatial reuse, however, requires aligning the beams of all nodes in a network. Aligning the beams is a difficult process which is complicated by indoor multipath, which can create interference, as well as by the inefficiency of carrier sense at detecting interference in directional links. This paper presents BounceNet, the first many-to-many millimeter wave beam alignment protocol that can exploit dense spatial reuse to allow many links to operate in parallel in a confined space and scale the wireless throughput with the number of clients. Results from three millimeter wave testbeds show that BounceNet can scale the throughput with the number of clients to deliver a total network data rate of more than 39 Gbps for 10 clients, which is up to $6.6\times$ higher than current 802.11 mmWave standards.

1 Introduction

Millimeter wave (mmWave) is emerging as the de facto technology for next generation wireless networks [24, 35]. The abundance of bandwidth available in mmWave frequencies (above 24 GHz) led to the design of wireless radios that can operate at several Gbps [2, 39, 56] and the wireless industry is constantly pushing towards incorporating these radios in wireless products [7, 8, 24, 25, 27, 50]. Hence, mmWave will significantly change the future of wireless LANs by delivering links at fiber-like speed. This will allow wireless LANs to handle the surge in IoT and mobile devices. Furthermore, it will enable new applications like multi-user wireless VR for education, professional training, and multiplayer games, where high bandwidth data must be streamed to each user in real-time [1, 11, 29]. It will also enable large scale robotic factory automation where many robots stream continuous real-time video back to servers that run AI algorithms and generate decisions to coordinate the robots [34, 57].

Enabling the above vision, however, requires scaling mmWave networks from a single communication link to a network of many links without compromising the throughput of each user. Fortunately, mmWave radios use very directional steerable narrow beams to focus their power. This presents a significant new opportunity for exploiting dense spatial reuse

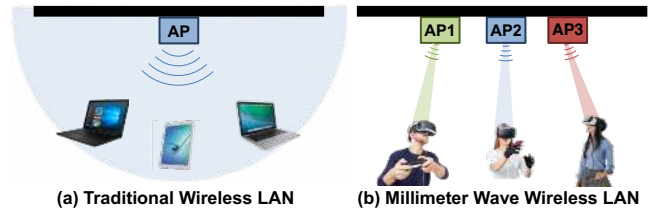


Figure 1: Spatial reuse in traditional WiFi vs mmWave networks.

to enable many links to simultaneously communicate at multi-Gbps data rates without interfering. Consider the example shown in Fig. 1. In the current broadcast model for 802.11 WLANs, whenever a node is transmitting, all other nodes must stay silent to avoid interference. With more users, the throughput is divided since the entire medium is shared. In contrast, the use of very narrow beams in mmWave networks allows several APs and clients to transmit and receive simultaneously on the same channel without interfering as shown in Fig. 1(b). Hence, mmWave can potentially scale the network throughput with the number of users by adding more APs.

The directional nature of communication, however, brings its own new challenges. Millimeter wave APs and clients need to align their narrow beams towards each other in order to communicate at very high data rates. Past mmWave research focused on developing algorithms and protocols to quickly find the best direction to align the beams for a single communication link [1, 23, 41, 49, 53, 64]. However, in a network with multiple links, selfishly choosing the best alignment for each AP-client pair independent of other APs and clients can create interference that severely harms the throughput of interfering links. First, due to multipath reflections, even if two nodes are transmitting in completely different directions, their packets might still collide. The problem is further complicated by the fact that carrier sense is ineffective at detecting interference since the narrow beams prevent mmWave radios from hearing nearby transmissions unless these transmissions are specifically directed towards them. Hence, we can rely on neither carrier sense nor the direction in which the nodes transmit to avoid interference.

In this paper, we introduce BounceNet, the first many-to-many millimeter wave beam alignment protocol that efficiently aligns the beams of many APs and clients in a manner that allows them to simultaneously communicate without interfering. To achieve this, we must address two key questions:

(1) How does BounceNet align the beams of all the APs

and clients in 3D space to densely pack as many links as possible? The challenge arises from the fact that the choice of beam alignment at any node is intertwined with the choices at other APs and clients. To address this, BounceNet leverages the sparsity in the mmWave channel. There is much past work that shows that mmWave signals travel along a small number of paths, e.g., 2 or 3 paths [5, 48]. This means that there is a small number of paths connecting any two nodes in the network. BounceNet leverages this sparsity to reformulate the problem as a signal level routing problem at the physical layer where wireless signals are routed along different “air paths” in a manner that avoids interference and maximizes network throughput. Routing physical signals is possible in mmWave due to the lack of scattering effects at such high frequencies which ensures the signal reflects off obstacles and does not scatter in many directions [48]. Hence, BounceNet can choose to route the signal along an isolated path by aligning the narrow beam towards that path.

By choosing a combination of direct and reflected paths to route the wireless signals, BounceNet can align the beams of all APs and clients in the network. While this allows it to maximize the number of links that can operate concurrently without interfering, it forces some APs and clients to communicate along reflected paths which typically achieve lower data rates. To address this issue, BounceNet generates several combinations of beam alignments and schedules them in different time slots; i.e., the transmissions of the links are routed along different paths in each time slot to ensure that each client gets high data rate while still maximizing the number of links that can operate simultaneously. BounceNet jointly solves the alignment and scheduling problems. We also model paths belonging to the same link as a supernode in a multi-layer conflict graph and weight them by the SNR of the path. This ensures that paths which deliver higher data rates are used more often as we describe in detail in section 6.

(2) *How does BounceNet quickly learn the paths and interference patterns in order to adapt the beam alignment in dynamic and mobile environments?* In dynamic environments, the propagation paths and the interference patterns constantly change. Thus, we must periodically perform a beam search to learn the directions of the paths along which an AP and client can communicate.¹ BounceNet must also learn the propagation paths that can result in interference between two links and, hence, needs to perform the beam search between all APs and clients in the network to learn all the possible paths. Past work has shown how to leverage sparsity to quickly learn the paths without scanning all directions and reduce the search time to a millisecond [23, 49]. However, for a network of N APs and clients, this process must be performed $O(N^2)$ times. For $N = 10$, even with fast algorithms like [23, 49], the overhead is 100 ms which is prohibitively expensive especially at multi-Gbps data rates.

¹Typically, the beam search is repeated every 100 ms in current standards like 802.11ad in order to track mobile users and maintain alignment.

Instead of performing the search independently for all APs and clients, BounceNet redesigns the beam search protocol to jointly find all the paths between the nodes. BounceNet coordinates the APs’ transmissions and then shares their measurements over the Ethernet which allows it to amortize the cost of the search and reduce it to $O(N)$. Since the beam search is inherent to mmWave and is required to maintain connectivity between clients and APs, BounceNet’s design does not introduce additional overhead compared to current standards. This allows BounceNet to quickly learn the paths and reconfigure the beam alignment to maintain high throughput as we describe in detail in section 5.

Implementation & Results: We have designed BounceNet to be backward compatible with the current mmWave wireless LAN standard 802.11ad/ay making it easy to integrate into future standards. Our design also addresses several practical challenges like side-lobe leakage from imperfect beam patterns and interference estimation. We have implemented BounceNet by using extensive real measurements from three indoor wireless testbeds:

- A 60 GHz testbed with 3° beam directional antennas.
- A 60 GHz testbed with 12° beam directional antennas.
- A 24 GHz testbed with 8-element phased arrays.

For a testbed with 10 APs and clients packed in an area of 860 sq.ft., our results show that BounceNet can scale the overall network data rate with the number of clients delivering over 39 Gbps for 10 clients. Furthermore, compared to the current 802.11ad standard that exploits spatial reuse, BounceNet can increase the average client throughput by 6.6×, 5×, and 3.1× for each of the above testbeds respectively. Compared to a baseline that aligns the beams of each link independent of other links, BounceNet increases the average client throughput by 1.27×, 2.7×, and 3.4× for each of the above testbeds respectively. BounceNet also improves the minimum data rate among all clients by up to 13.5× compared to the baseline which can create interference that severely harms some clients. Finally, Fig. 2 shows an example snapshot of a time slot where BounceNet exploits multipath to enable all 10 APs and clients, in the 60 GHz testbed with 12° beams, to communicate at the same time without interfering, hence demonstrating BounceNet’s ability to enable extreme spatial reuse.

Contributions: We make the following contributions:

- We present the first many-to-many beam alignment protocol that can efficiently align the beams of a network of APs and clients to maximize the number of links that can operate concurrently.
- We demonstrate the opportunity of routing physical signals along different paths that bounce off the environment to improve the spatial reuse of the network. We harness this opportunity to design new algorithms that maximize network throughput while maintaining a lower bound of fairness for each client.

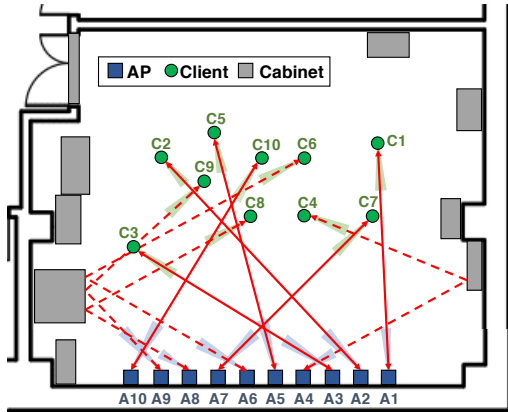


Figure 2: Example of BounceNet’s signal routing in practice.

- We extensively evaluate our system through micro-benchmark measurements, trace-driven simulations, and experiments using 3 testbeds. Our results demonstrate the first design of a wireless LAN that can deliver more than 39 Gbps to 10 clients.

2 Related Work

Millimeter Wave Networks: BounceNet is related to recent work on increasing the speed and robustness of beam alignment in mmWave networks to enable mobility [19, 20, 23, 41, 42, 49, 53, 63–65] and avoid blockage [1, 31, 32, 41, 52, 54, 61]. All this work, however, focuses on a single communication link. BounceNet is the first to demonstrate many-to-many beam alignment. It is complementary to these systems and can benefit from faster beam search to discover the paths between nodes.

BounceNet also builds on past work in mmWave that uses 60 GHz wireless links in data centers [12, 21, 66] and leverages reflections off the ceiling to improve the throughput and avoid blockage [12]. Data centers, however, have static and known topologies with predictable interference models [12], and this does not hold in 802.11 LANs where the clients can move.

Our work is also related to recent mmWave work that deploys multiple APs to deal with blockage [60, 62]. [60] leverages multiple APs and allows clients to switch between them whenever blockage occurs in VR applications. However, it requires brute-force training to map all reflectors in the environment and relies on sensors in VR headsets to track the direction of users. [62] addresses blockage by having multiple APs jointly transmit the same signal to the clients. However, the method works only for downlink traffic and requires phase and frequency synchronization to ensure the signals sum up coherently. Achieving such level of synchronization is difficult and adds significant complexity to the design [22, 46]. BounceNet opts for a simpler design that scales the throughput of the network for both downlink and uplink traffic without

requiring phase, frequency or packet level synchronization. It also learns the reflected paths in real-time.

Some recent simulation-based work for mmWave wireless PANs (Personal Area Networks) [3, 4, 17, 18, 44, 58] and mmWave mesh networks [38] tries to exploit spatial reuse. However, these solutions assume that the exact locations of the nodes are known a priori and can be used to compute the interference between links while ignoring multipath. BounceNet, on the other hand, designs and empirically tests a system that can work in the presence of multipath without prior assumptions of the clients’ locations.

Finally, [14, 16] use MU-MIMO in mmWave and demonstrate concurrent transmissions to two clients from one MU-MIMO AP. BounceNet’s beam alignment algorithm is complementary to MU-MIMO and can benefit from having APs that support MU-MIMO to further scale the gains.

Enterprise WiFi and WLANs with Directional Antennas:

Past work has designed protocols for mobile ad-hoc networks and WLANs with directional antennas [9, 10, 28, 33]. However, past work can support only large cone beams (e.g. 45° and 60° cones) at data rates of at most tens of Mbps. The scale of the problem is far more extreme in mmWave with narrow pencil beams of few degrees to sub-degree beamwidth at data rates of multi-Gbps. Hence, the overhead of past protocols can be prohibitively expensive in mmWave. Moreover, most of these protocols assume the locations of the nodes are known and ignore multipath [9, 10, 28].

The closest to our work is [33] which leverages directional phased arrays at 2.4 GHz to increase spatial reuse. However, [33] assumes only APs to have directional antennas which simplifies the problem since the clients can easily perform interference detection in the omnidirectional mode. Furthermore, the scheduling algorithm in [33] is exponential in the number of APs and hence is only shown to work for 3.

Past work had designed centralized scheduling algorithms for enterprise WiFi networks [51]. However, WiFi networks are omni-directional. Extending past algorithms to deal with directionality is non-trivial since the interference or conflict graph used for scheduling is itself dependent on the choices of beam alignment and there is a combinatorial number of choices as we discuss in section 5. BounceNet jointly solves the beam alignment and scheduling problems to deliver an efficient algorithm.

3 Background

BounceNet is designed to be backward compatible with 802.11 millimeter wave standards for indoor wireless LANs. In this section, we provide a brief overview of the 802.11ad standard for 60 GHz networks [26, 40].²

²Note that another standard in the works is 802.11ay. However, it fully inherits the same PHY and MAC structure of 802.11ad. The main difference is the introduction of MIMO [15].

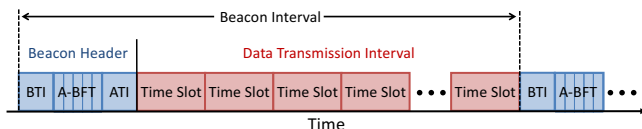


Figure 3: 802.11ad/ay Beacon Interval Structure.

The standards divide time into transmission cycles typically referred to as Beacon Intervals (BI) which consist of two phases shown in Fig. 3. The first is the association phase which is referred to as the Beacon Header Interval (BHI). It is used to associate the clients with the AP and perform beam alignment. The second is the transmission phase which is referred to as Data Transmission Interval (DTI) where time slots are allocated for communication between the AP and associated clients. We will first describe these phases for the case of a single AP and multiple clients. We will then extend our description to multiple APs.

A. Association Phase:

The beacon header shown in Fig. 3 is used to associate the clients with the AP and perform beam alignment so that both the clients and the AP know which direction they should point their beam during data transmission.

The beacon header starts with a Beacon Transmission Interval (BTI) where the AP transmits announcement frames in all directions by sequentially sweeping its narrow beam along different sectors. During this time, the clients listen to the channel in all directions using a quasi-omnidirectional beam pattern so that they can receive packets from all paths. The announcement frames are marked with the sector ID along which they are sent allowing each client to discover the directions which the AP can use to send it data packets.

BTI is then followed by Association Beamform Training (A-BFT) which reverses the above operation. The AP uses a quasi-omnidirectional beam pattern so that it can hear clients from all directions while the clients sweep their narrow beam along different sectors. This allows the AP to discover the beam directions which the client can use to send its data packets and send it feedback to inform it of these directions. A-BFT is divided into multiple slots. Each client selects a random slot to perform its sweep. If two clients collide in an A-BFT slot, they will not get feedback from the AP and they can try again in another random slot.

The above process enables the AP and client to align their beams towards each other so that they can boost their SNR and use very high data rates for data transmission. However, during this association phase and before aligning their beams, the AP and clients use a control PHY with a low data rate of 27.5 Mbps to ensure the frames can be decoded correctly at low SNR. The beacon header finally ends with Announcement Transmission Interval (ATI), where the AP and associated clients exchange control frames such as information regarding time slots that have already been allocated to the client.

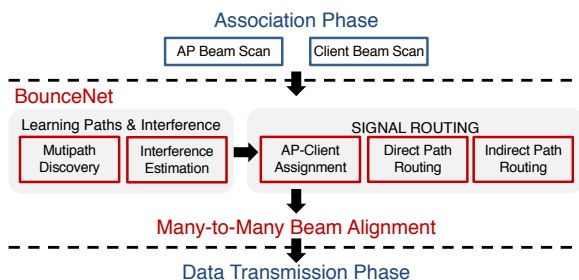


Figure 4: BounceNet's System Architecture.

B. Transmission Phase:

The data transmission interval (DTI) is divided into time slots. The AP either uses TDMA to allocate each slot to a certain client or it allows the clients to contend for each time slot using CSMA. CSMA, however, does not work for directional networks [9, 33]. Hence, TDMA is more commonly used especially for video streaming applications where clients require dedicated slots in every beacon interval to ensure high quality and reliability.

For data transmission, the standard provides 32 different modulation and coding schemes (MCS) including single carrier modulation and OFDM modulation. Commercial products, however, adopt single carrier modulation due to the high power consumption of OFDM [45, 55]. Hence, in this paper, we will focus on single carrier: MCS1 to MCS12 which provide data rates between 385 Mbps and 4.62 Gbps [26].

C. Multiple Access Points:

In the case of multiple APs, a lead AP is selected. The lead AP divides the beacon interval into smaller beacon intervals called beacon service periods (BSP). Each BSP has its own beacon header and data transmission period and it is allocated to one AP. All other APs must stay silent during this service period. In order to enable spatial reuse, the lead AP can allocate a service period to two APs and request that they measure mutual interference and report back. If no interference occurs, it allocates the same service period to these APs in subsequent beacon intervals. Unfortunately, our results show that such a greedy mechanism for exploiting spatial reuse is unable to scale the network throughput with the number of clients.

4 BounceNet Overview

BounceNet's goal is to align the beams of all APs and clients in the network in a manner that maximizes spatial reuse. This allows WLANs to add additional APs to quickly scale their throughput with the number of clients.

We have designed the BounceNet protocol to support *independent flows*. This means that for an AP-client pair that is assigned to communicate along a path in a time slot, its link flow runs independently of other links for that time slot. The AP and client can transmit packets on the uplink or downlink without interfering with other links. The pair does not have to share any data packets or synchronize the individual packet transmissions with other APs or clients.

BounceNet is also backward compatible with 802.11ad/ay. It maintains the same high-level structure. BounceNet’s architectural flow is shown in Fig. 4. It uses a controller that sits between the association phase and the data transmission phase of the protocol. BounceNet uses association phase to learn the paths and interference in the network and then runs its signal routing algorithm which dictates the many-to-many beam alignment in the data transmission phase.

BounceNet starts with an association phase similar to 802.11 where the APs and clients sweep their beams to collect information about the directions in which their signals can reach other APs and clients. This information is then aggregated at the APs, and fed to the BounceNet controller which allows it to discover all the paths connecting any two nodes in the network. We refer to this as multipath discovery (Section 5.1). BounceNet then uses the phased array beam patterns and the learned paths to estimate the interference created by routing signals along each path (Section 5.2).

BounceNet uses the results to route physical signals along propagation paths in a manner that maximizes the number of AP-clients pairs that can communicate simultaneously. Ideally, we would have liked to treat all APs as one large AP with many paths to all clients and find the optimal routing. However, this significantly increases the complexity of the problem and will require very fast handoff between APs to allow clients to switch APs within a beacon interval.³ Hence, BounceNet assigns a single AP to each client for communication during the entire Beacon Interval.

To reduce the complexity of the system and ensure fairness, BounceNet performs signal routing in three stages:

- **Stage 1:** Associate each client to communicate with one AP for the duration of the entire beacon interval. (Section 6.1)
- **Stage 2:** Route the signal of each AP-client pair along their direct or highest throughput path in a manner that maximizes the number of links that can communicate in a given time slot without interfering. (Section 6.2)
- **Stage 3:** Route additional signals of AP-client pairs along their indirect paths to increase throughput without interfering with existing transmissions. (Section 6.3)

The above signal routing results in several beam alignments that are used for transmissions between APs and clients during each time slot of the data transmission phase. The entire process is repeated every beacon interval to adapt to changes in the environment and accommodate client mobility.

5 Learning Paths & Interference

BounceNet must first map all the paths between all nodes in the network and discover the potential interference between paths. Typically, for a network with N APs and N clients, this would require collecting $O(N^2)$ measurements. BounceNet

³Such fast handoffs are not feasible in mmWave networks because they require transferring the buffer at one AP to another AP at the time scale of few ms which would overwhelm the backhaul.

Algorithm 1 BounceNet Multipath Discovery

```

 $N \leftarrow$  Number of APs
 $\forall$  Clients  $\rightarrow$  Set quasi-omnidirectional beam
 $\forall$  APs  $\rightarrow$  Set quasi-omnidirectional beam
Begin BTI:
for  $m \in \{1, \dots, N\}$  do
     $AP(m) \rightarrow$  Set directional beam
    for  $\theta \in$  Sectors do
         $AP(m) \rightarrow$  Transmit frame in direction  $\theta$ 
         $\forall$  Clients & APs
        if Frame Received then
            Paths. $AP(m)\{\theta\} \leftarrow SNR$ 
         $AP(m) \rightarrow$  Set quasi-omnidirectional beam
Begin A-BFT:
Repeat the above process for clients.
Report Paths back to APs in transmitted frames.

```

instead redesigns the 802.11ad/ay protocol and exploits its beam alignment phase to extract all the paths from $O(N)$ measurements that are already part of the standard protocol.

5.1 Multipath Discovery

As described earlier, in case of multiple APs, the current standard divides the beacon interval into smaller beacon intervals and dedicates each interval to one AP. Instead, BounceNet aggregates them into one beacon interval with one beacon header and one data transmission interval. In particular, BounceNet only expands the BTI, shown in Fig. 3, to allow all APs to perform their beam scan of sequentially sweeping all sectors. While an AP is performing a sweep, all other clients and APs set their antenna to a quasi-omnidirectional mode and record the sector IDs of the frames they receive along with the SNR of the signals. A-BFT is then performed by assigning each client to a slot. While some client is performing its sweep, all other clients and APs set their beam to quasi-omnidirectional and record the sector IDs and SNRs of the frames received from the client. Algorithm 1 shows pseudocode for BounceNet’s association phase.

The above process recovers a list of directions from which any node (AP or client) in the network can reach any other node. However, this might not be sufficient for discovering the paths between an AP and a client. Consider the example shown in Fig. 5(a) where there are three paths between an AP and a client. During BTI, we discover that the AP can reach the client by transmitting in one of three directions: 30° , 60° or 150° as shown in Fig. 5(b). During A-BFT, we discover that the client can reach the AP by transmitting in one of three directions: 30° , 110° or 150° as shown in Fig. 5(c). Unfortunately, since we do not know the position and orientation of the client, we do not know which direction at the AP corresponds to which direction at the client.

To address this, BounceNet needs to match the directions corresponding to the same paths by correlating the SNRs

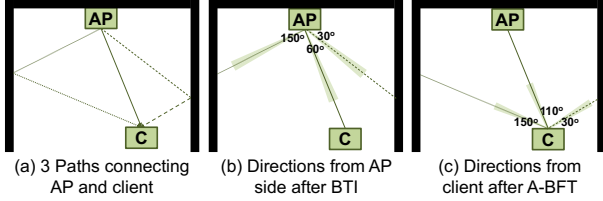


Figure 5: Multipath Discovery in BounceNet.

recorded from the client side and from the AP side. For instance, the directions corresponding to the direct path can be easily identified since typically the direct path delivers significantly higher SNR compared to indirect paths as we empirically show in Fig. 14(a) in section 8. However, in some cases, there could be two indirect paths that show similar SNR values (within 1 dB of each other). In such situations, correlation might lead to erroneous matching due to the inherent noise in SNR measurements. Fortunately though, as we show in section 8, the number of reflected paths between a pair of nodes in millimeter wave is quite small, e.g. 1 to 2 paths [5, 48]. Hence, at most, only two paths would remain ambiguous after the correlation step. BounceNet can then leverage the beam refinement option in 802.11ad which allows AP-client pairs to test pairwise directions to resolve such ambiguity. This incurs four more measurements. However, these measurements are taken while both AP and client beams are directional. Hence, they are transmitted at high data rate and incur negligible overhead.

5.2 Interference Estimation

Once we have discovered all the paths between the nodes in the network, we can estimate the interference caused by using any two paths simultaneously. BounceNet defines interference between paths as opposed to between nodes. If two paths interfere, then signals cannot be simultaneously routed along these two paths. We would like to keep the flows independent and avoid synchronization. Hence, at any point in time, both paths can be used to transmit uplink traffic, downlink traffic, or one path is used on the uplink while the other is used on the downlink. Consider a path between AP 1 and client 1 and another path between AP 2 and client 2 as shown in Fig. 6. Interference can occur in one of four cases: between AP 1 and AP 2, client 1 and client 2, AP 1 and client 2, or AP 2 and client 1 if there is a path connecting any of these pairs.

Formally, each path is defined by two angles corresponding to the direction from which it leaves one node and arrives at another node. We distinguish two types of paths:

- *Communications Paths*: defined as $(\theta_{AP_i}, \theta_{C_i})$ between AP 1 and client 1 as well as between AP 2 and client 2.
- *Interference Paths*: defined as $(\phi_{AP_i}, \phi_{C_j})$ between AP 1 and client 2 or AP 2 and client 1. They can also be defined as $(\phi_{AP_i}, \phi_{AP_j})$ or (ϕ_{C_i}, ϕ_{C_j}) .

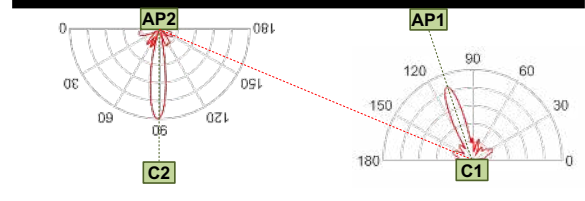


Figure 6: Estimating Interference using phased array beam patterns.

Ideally, it would be sufficient to check the directions of the paths to discover if interference occurs. Suppose AP 1 and client 1 can communicate along the path $(\theta_{AP1}, \theta_{C1})$ and AP 2 and client 2 communicate along the path $(\theta_{AP2}, \theta_{C2})$. In this case, for example, AP 2 will create interference at client 1 only if there exists an interference path (ϕ_{AP2}, ϕ_{C1}) where ϕ_{AP2} is in the direction of θ_{AP2} and ϕ_{C1} is in the direction of θ_{C1} . A similar rule can be used to detect interference between the other pairs.

Unfortunately, such a simple interference detection scheme will not work in practice. This is because the antenna beam patterns are not ideal cones. They have side lobes and can leak signal in other directions. Consider the example in Fig. 6, while AP 2 is transmitting in direction $\theta_{AP2} = 90^\circ$, its signal might leak along another direction $\phi_{AP2} = 160^\circ$ and reach client 1. To address this, BounceNet incorporates the phased array transmit and receive beam patterns into its interference estimation.⁴ Specifically, to estimate interference between any pair of nodes, we consider all the interference paths between the two nodes and weight them by the beam pattern gains. Formally, when AP 2 directs its beam towards client 2 in the direction θ_{AP2} , it will have a beam pattern of $B_{\theta_{AP2}}(\phi)$. Similarly, when client 1 directs its beam towards AP 1 in the direction θ_{C1} , it will have a beam pattern of $B_{\theta_{C1}}(\phi)$. The interference created by AP 2 on client 1 due to an interference path $P = (\phi_{AP2}, \phi_{C1})$ can be calculated as:

$$B_{\theta_{AP2}}(\phi_{AP2}) \cdot B_{\theta_{C1}}(\phi_{C1}) \cdot SNR(P)$$

where $SNR(P)$ is the normalized SNR⁵ of the path P from AP 2 to client 1 measured during multipath discovery.

The maximum interference AP 2 causes can then be estimated as the constructive sum of leakage along all paths between AP 2 and client 1:

$$INR = \sum_{P=(\phi_{AP2}, \phi_{C1})} B_{\theta_{AP2}}(\phi_{AP2}) \cdot B_{\theta_{C1}}(\phi_{C1}) \cdot SNR(P)$$

where INR is the interference-to-noise ratio. BounceNet repeats this estimation eight times: from AP 1 to AP 2 and client 2, from AP 2 to AP 1 and client 1, from client 1 to AP 2 and

⁴Such patterns can be modeled or measured to account for imperfections in the mmWave phased arrays.

⁵The SNR is normalized by the antenna beam patterns used during the measurement of the SNR value in the multipath discovery phase.

client 2 and from client 2 to AP 1 and client 1. BounceNet then defines the INR between the two communication paths as the maximum INR of all these 8 values.

Two points are worth noting:

- The above interference estimation does not assume to know the location or orientation of the APs or the clients. It also does not rely on knowing the room geometry or the use of ray tracing. It only requires the direction of the propagation paths (ϕ_1, ϕ_2) between nodes in the network and the associated signal strength along the paths.
- BounceNet is able to constantly maintain an up-to-date view of the multipath and interference pattern in the network since it obtains fresh measurements from the AP and client sweeps at the start of every Beacon Interval (which is approximately 100 ms). This feature allows BounceNet to deal with dynamic network conditions and accommodate for client mobility.

6 BounceNet’s Signal Routing

Once BounceNet knows all the paths connecting the nodes and all the interference between the paths, it can route signals to/from clients in a manner that maximizes the number of AP-client pairs that can communicate in parallel. The choice of routing will govern the many-to-many beam alignment. BounceNet simplifies the problem by dividing it into three stages: *AP-Client Association*, *Direct Path Routing*, and *Indirect Path Routing*. We will elaborate on each stage below.

6.1 AP-Client Association

In the first stage, our goal is to associate each client to one AP for communication during the subsequent Data Transmission Phase of the Beacon Interval. Each client can associate with one AP, whereas each AP can serve multiple clients. Hence, for a network with N APs and N clients, we have N^N possible assignments. Trying all assignments is computationally infeasible. Thus, we develop an algorithm that sequentially assigns the clients to APs, with the objective of increasing throughput while minimizing the interference in the network. The intuition behind our algorithm is based on the following observations:

- In indoor settings, clients can typically achieve the highest data rate if they have a direct line-of-sight path to an AP. Hence, to ensure fairness, we should assign each client to an AP with a direct line-of-sight path.
- To maximize spatial reuse and throughput, we should avoid assigning multiple clients to the same AP unless the client cannot find any unassigned AP with a direct path.

Our algorithm works as follows. For each client, BounceNet keeps a list of best APs which have a direct path (high SNR path) to that client. BounceNet starts with the client with the least number of best APs and assigns it to one of the APs in its best AP list. It then adds this AP-Client pair

to a list of already assigned links. For every subsequent client, BounceNet finds an AP from its best AP list such that: (1) the AP has not yet been assigned to a client, and (2) when communicating along their direct path, the AP-Client pair creates the minimum amount of interference on the direct paths of the already assigned links.⁶ If no such AP exists, BounceNet simply picks the AP from the client’s best AP list that creates the least interference.

The above algorithm is a best effort algorithm to assign each client to an AP with a direct path that creates the least amount of interference between the links. In the worst case, the best AP list of each client contains N APs. Then, while assigning the i^{th} client, BounceNet must compute the interference created by choosing one of the $N - i$ remaining APs on the i assigned links. Hence, the complexity is: $\sum_{i=1}^N (N - i)i = O(N^3)$. This reduces the complexity from exponential $O(N^N)$ to polynomial $O(N^3)$.

6.2 Direct Path Routing

Once each client is assigned to an AP, we will have N unique direct paths. BounceNet starts by routing signals to/from clients along these direct paths. Decoupling the signal routing along the direct and reflected paths simplifies the problem and allows us to ensure fairness among links when it comes to routing signals through their highest throughput paths, i.e. their direct paths. In the next section, we will show how BounceNet routes additional signals along indirect paths to enhance throughput.

A. Scheduling of Direct Paths

BounceNet uses graphs to solve the problem. It starts by building the *Direct Path Conflict Graph*: $G(V, E)$. V represents the set of vertices in the graph. Each vertex v corresponds to a direct path between an AP-client pair. E represents the set of edges in the graph. An edge $e_{u,v}$ exists between vertices u and v if the corresponding paths interfere. We use the estimation from section 5.2 to compute the interference between paths, and if the $INR > 0$ dB, we assign the paths as interfering.

In each time slot, BounceNet’s goal is to schedule routing signals along as many paths as possible. Traditionally scheduling is modeled and solved as a minimum graph coloring problem on the conflict graph [30, 36, 47, 59]. This finds the minimum number of colors required to color the graph such that no two vertices connected by an edge share the same color. Thus, paths corresponding to vertices of the same color can be scheduled and used concurrently in the same time slot. This will minimize the number of time slots needed to schedule the paths while ensuring that each path gets one time slot to route signal to/from the client. Fig. 7(a) shows a possible minimum coloring of a graph which requires 3 colors. This

⁶The amount of interference is estimated as the sum of the INRs computed in Section 5.2.

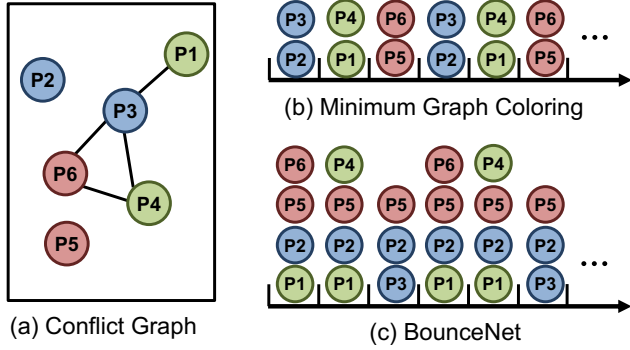


Figure 7: Scheduling of Direct Paths.

means that we can schedule all paths within 3 time slots as shown in Fig. 7(b). Since there are 6 paths, this will give $2 \times$ higher throughput than a scheduling which does not utilize spatial reuse and routes signals only along one path at any point in time.

B. Fairness in Millimeter Wave Networks

The above formulation can leverage spatial reuse to increase throughput while ensuring that each client gets an equal share of the time on the channel. This notion of fairness, however, is suboptimal in mmWave networks and needlessly wastes throughput. Due to the use of very directional beams in mmWave networks, the medium is no longer “equally” shared among all clients. Consider the example in Fig. 7(a). Paths 2 and 5 do not interfere with any other path and hence we should route signals through these paths in every time slot. Not doing so would reduce the throughput without benefiting anyone in the network. On the other hand, paths 4 and 6 share their medium with two other paths since they interfere with two other paths. Hence, a path should get a share of the medium which is at least a fraction of the number of paths it shares its medium with. For example, we should route signals through paths 4 and 6 in $1/3$ of the time slots, whereas we should route signals through paths 2 and 5 in all time slots since they interfere with no one.

Formally, if a path interferes with d other paths, it shares its medium with these d paths and hence should get a share of at least $1/(d+1)$. In the conflict graph G , d will correspond to the degree of the vertex, i.e. the number of edges that the vertex has. Using this new notion of fairness, we develop an algorithm to route signals through direct paths in a manner that achieves higher throughput while maintaining fairness.

C. BounceNet’s Algorithm

BounceNet starts by trying to maximize the number of paths that can be used in each time slot. Maximizing the number of paths is theoretically equivalent to solving a maximum independent set problem. The maximum independent set refers to the maximum number of vertices that do not share any edges. For example, in Fig. 7(a), the maximum indepen-

Algorithm 2 BounceNet Scheduling of Direct Paths

```

 $G(V, E) \leftarrow$  Direct Path Conflict Graph
 $M \leftarrow$  Number of time slots in beacon interval
 $F_1(u) = M \forall u \in V$ 
for  $t \in \{1, \dots, M\}$  do
   $W_t \leftarrow$  WEIGHTEDMAXINDEPENDENTSET( $G, F_t$ )
  for  $u \in W_t$  do
    if  $F_t(u) > 2(d(u) + 1)$  then
       $F_{t+1}(u) = F_t(u) - (d(u) + 1)$ 
    else
       $F_{t+1}(u) = 0$ 

```

dent set can be formed of paths 1, 2, 4, and 5 since none of these paths share edges, i.e. none of them interfere. Routing signals through these paths in every time slot will achieve the highest possible throughput. However, it will result in starvation of some clients whose paths are never included in the maximum independent set, e.g. Path 3 in Fig. 7(a).

Instead, BounceNet uses a variant of the same problem referred to as the Weighted Maximum Independent Set. The idea is to give each vertex u a weight $F(u) \geq 0$. We then find the set of vertices W that maximize the sum of weights such that no two vertices in W share an edge. More formally, we find the set W that satisfies:

$$\text{maximize } \sum_{u \in W} F(u) \text{ such that } \forall u, v \in W, e_{u,v} \notin E \quad (1)$$

BounceNet solves the above optimization problem for every time slot and schedules to route paths corresponding to the vertices in W to each of the time slots. After each time slot, BounceNet decrements the weights of each of the vertices in W by an amount proportional to the interference it creates in the network, i.e. the degree of the vertex d . Hence, if we initialize all the weights equally, then for the first time slot, BounceNet will pick a Maximum Independent Set. However, as the algorithm proceeds, the weights of the scheduled paths keep getting decremented, and eventually paths that interfere with the paths in the Maximum Independent Set start to get picked in W , and in turn get scheduled.

Pseudocode of this algorithm is shown in Algorithm 2. Fig. 7(c) shows an example of the output of BounceNet’s direct path routing. In this example, BounceNet’s algorithm achieves $3.66 \times$ higher throughput while ensuring fairness, i.e. each path gets scheduled at least $1/(d+1)$ of the time.

D. Analysis

If BounceNet wishes to schedule the nodes into M slots, it initializes all the weights to M . Then, every time a vertex u is picked, its weight is decremented by $d(u) + 1$ where $d(u)$ is the degree of this vertex. After this vertex has been picked up $M/(d(u) + 1)$ times, its weight becomes 0. Once the weight of a vertex becomes zero, its inclusion in W can no longer help maximize the sum of weights, and hence it does not get

picked up (or in our context, the path is no longer used) after that. However, by the time the weight of the vertex reaches 0, it has already been scheduled in $1/(d(u) + 1)$ of the time slots and hence fairness is achieved. For example, if a vertex has degree $d = 0$, i.e. it does not interfere with anyone, it will be picked up every time since it will always help maximize the sum of weights. Every time it is picked, its weight is decremented by 1. Its weight will reach 0 only after it has been scheduled M times which means it has been scheduled in all time slots. In Appendix A, we prove the following lemma: **Lemma 6.1** *If $t = O(M \log(NM))$, then $F_t(u) = 0 \forall u \in V$*

Algorithm 2, however, requires solving a Weighted Maximum Independent Set problem which is NP-hard [13]. This would require an exponential time algorithm to find the optimal solution, which would be infeasible for any real-time implementation. We use the approximation algorithm from [13] to solve this problem. Empirically we find that the algorithm is at most two timeslots worse than optimal. However, in many cases, the algorithm achieves the optimal. This is because the sparsity renders the *Direct Path Conflict Graphs* in mmWave networks as chordal with very high probability. Chordal graphs are graphs in which all cycles of four or more vertices have a *chord*. For such graphs, [13] is optimal.

6.3 Indirect Path Routing

In this section, we will show how BounceNet will route additional signals along indirect multipath routes to increase the throughput without creating interference to signals being routed along the direct path.

BounceNet’s indirect path routing is best understood through an example. Let us consider the direct path scheduling result shown in Fig. 7(c). During the first time slot, paths 1, 2, 5 and 6 were scheduled. Hence, clients 1, 2, 5 and 6 can communicate on their direct paths during this time slot. Note that a client can route its signal through only one path during any time slot. As a result, we only need to consider whether we can route signals through multipath for clients 3 and 4.

To this end, BounceNet forms an *Indirect Path Conflict Graph*. This graph includes vertices corresponding to the direct paths that have been scheduled as well as vertices corresponding to indirect paths of AP-client pairs that have not been scheduled in this time slot. Fig. 8(a) shows an example of this graph where client 3 has two indirect paths to its AP and client 4 has three indirect paths to its AP. Indirect path vertices corresponding to the same client are always in conflict since the client can use only one of those indirect paths. Hence, vertices corresponding to indirect paths of the same client form a fully connected subgraph which we will refer to as a *supernode*. We then estimate the interference that the indirect paths can create on direct paths that are already scheduled as well as other indirect paths.

Direct paths have already been scheduled and hence they are locked. Any indirect path that interferes with the direct

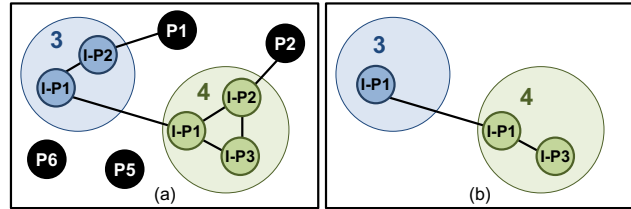


Figure 8: Indirect Path Conflict Graph before & after pruning.

path cannot be used in this time slot and hence can be eliminated from the indirect path conflict graph. Thus, BounceNet prunes the graph by removing all vertices that interfere with direct paths as well as vertices corresponding to direct paths as shown in Fig. 8(b). The resulting graph is typically much smaller and formed only of supernodes and vertices corresponding to indirect paths. BounceNet can route signals through any of the remaining indirect paths without interfering with signals being routed through the direct paths.

In order to schedule indirect paths, BounceNet uses the same algorithm as before where it maximizes throughput by solving a maximum weighted independent set problem on the *Indirect Path Conflict Graph*. However, BounceNet has to take into account two key differences:

- Unlike direct paths where there is small variance in SNR, the SNR of indirect paths can vary significantly as we will show in section 8. Hence, BounceNet should give indirect paths with higher SNR more weight. To do so, BounceNet gives each *supernode* a weight of M and divides this weight among its indirect path vertices in a manner proportional to the data rate that each indirect path can achieve. For example, if *supernode* 4 in Fig. 8 has indirect paths with SNRs 3 dB, 5 dB, and 7 dB, then it can deliver data rates of around 1.1 Gbps, 1.9 Gbps, and 2.5 Gbps respectively. Hence, its indirect paths will be weighted as $0.2M$, $0.35M$, and $0.45M$. This ensures that the higher data rate paths have a higher chance of getting picked.
- The degree d of a vertex no longer corresponds to the number of other clients it shares the medium with since vertices of the same *supernode* belong to the same client. Hence, instead of decrementing the weight of the node by $d + 1$, we decrement it by $d - s + 1$ where s is the number of other vertices that remain in the supernode after pruning the graph. For example, in Fig. 8(b) the indirect path in *supernode* 3 has $s = 0$ whereas in *supernode* 4 have $s = 1$.

7 Testbed and Implementation

We evaluated BounceNet using three indoor testbeds that operate at 60 GHz and 24 GHz. The 60 GHz testbeds used Pasternack PEM009 radios [43] shown in Fig 10(a). One testbed is equipped with directional antennas with beamwidth 3° and the other with 12° antennas shown in Fig. 10(b). The 60 GHz Pasternack modules are connected to USRP software defined radios through a Balun circuit to sample the signal.



Figure 9: **Indoor Experimental Space:** (a) Lecture Hall (b) Atrium (c) Lounge (d) Empty Room (e) Lab (f) Office Space.

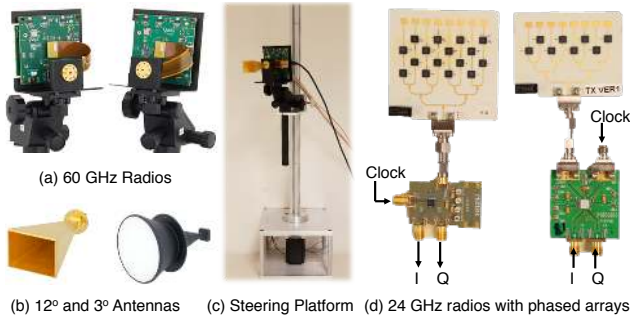


Figure 10: Experimental hardware used to evaluate BounceNet.

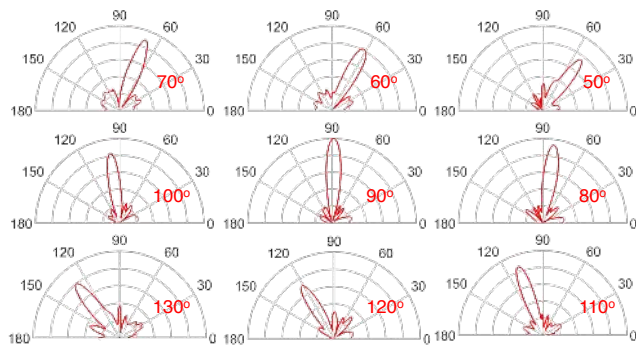


Figure 11: Example beam patterns of the 24 GHz phased arrays.

They are also mounted on a steerable platform shown in Fig 10(c) controlled through an Arduino.

The 24 GHz testbed used two radios, each equipped with an 8-element phased array shown in Fig. 10(d). The radios use HMC815B and HMC977 IQ up/down converters from Analog Devices which operate between 21 GHz and 27 GHz with 3.75 GHz of bandwidth. The integrated boards shown in Fig. 10(d) also include RF amplifiers and a frequency doubler. The boards are fed a clock in the range 10.5 GHz to 14.5 GHz from a TI LMX2594 PLL which is doubled to the 24 GHz range. The I and Q signals are connected to USRP software defined radios where the signals are collected. Fig. 11 shows examples of the beam patterns of the phased array that we obtain from our own empirical measurements. Note that while the beam patterns from some commercial phased arrays have much larger side lobes, we are able to achieve beam patterns as shown in Fig. 11 by leveraging the online algorithm for phased array calibration presented in [37].

We use the Tektronix DPS77004SX oscilloscope which

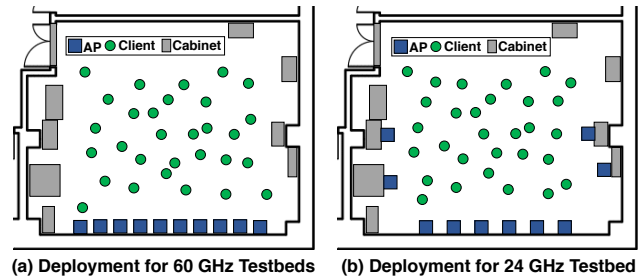


Figure 12: Placement of APs in the 60 GHz and 24 GHz testbeds.

samples at 200 GS/s and has a bandwidth of 70 GHz to calibrate the transmitted power of both 60 GHz and 24 GHz radios to match FCC regulations. We also use it to calibrate the measured power and noise floor of the USRPs.

Due to the large overhead of real-time processing and the limited bandwidth of USRPs, we use the software radios to measure interference and signal-to-noise ratio, which we map to the minimum achievable data rate using the receiver sensitivity table of 802.11ad [26] with 1% packet loss rate. We then used these testbed measurements to run trace-driven simulations using an 802.11ad ns3 library that takes phased array beam patterns into account [6]. We also modified this library to implement BounceNet. We then empirically verified the results by testing the interference and making sure any pair of paths used in a given time slot does not interfere. We then report the data rates per client as well as the overall network data rate. Finally, we also study the impact of our system when integrated with higher layer protocols like TCP and UDP and report application level throughput results.

We collected measurements in different rooms in order to evaluate the level of multipath and verify that BounceNet can exploit this multipath to maximize the number of links. We tested in six different types of rooms shown in Fig. 9: a lecture hall, an atrium, a lounge, a completely empty room, a lab space, and an office space. The full BounceNet protocol was evaluated in the lab which is 860 sq.ft. of space. The APs were deployed along the walls of the lab with the clients scattered across the room as shown in Fig. 12. We vary the number of APs and clients from 1 to 10. In every run, the clients are assigned randomly to these locations. We tested 5000 different configurations of locations. To emulate mobility, we move the clients in 5 cm steps along a path where we run scans and collect measurements for each step in the path.

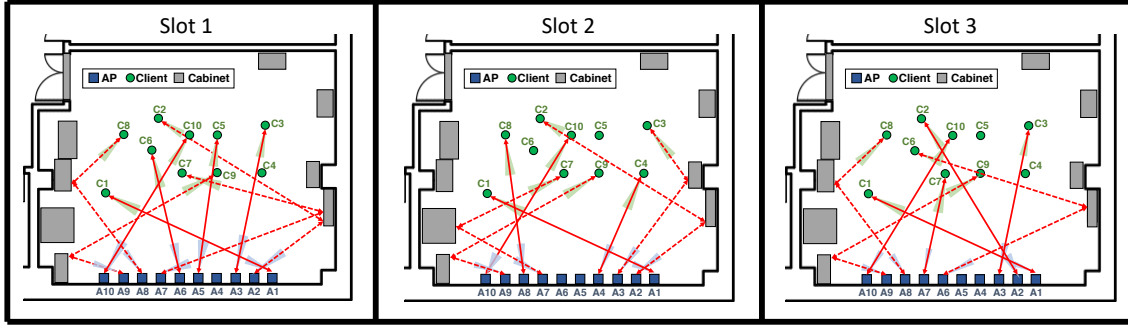


Figure 13: Beam Alignments computed by BounceNet for 12° beam testbed.

8 Microbenchmark Results

We start our evaluation with a few microbenchmarks that provide insights into the working of the system as well as the characteristics of mmWave networks before we present the evaluation results.

A. Multipath in mmWave Networks:

BounceNet leverages multipath in mmWave networks to maximize the number of links that can operate at the same time. Table 1 shows the distribution of the number of reflected multipath per link in each of the six rooms shown in Fig. 9. The results show that for all rooms except the atrium, in about 80% of the cases the client has 1 to 2 reflected paths through which it can route its signal to the AP. This is expected as the atrium is a large open space with limited reflectors. The results also show that very few clients see 3 or 4 indirect paths due to sparsity in mmWave.

Fig. 14(a) shows the CDF of the SNRs of the direct and reflected paths respectively measured from our testbeds. We observe that direct paths always provide sufficient SNR to support the highest data rate of 4.62 Gbps. The variation in direct path SNRs is small and the median SNR of direct paths is 15 dB larger than the median SNR of reflected paths which motivates BounceNet’s design to split routing signals along direct and indirect paths into two stages. Furthermore, the SNRs of indirect paths can vary between 5 dB to 20 dB and hence it is important to take the SNR of indirect paths into account when deciding which indirect path to route signals through as we have described in section 6.3.

B. Accuracy of Interference Estimation:

Here, we evaluate the accuracy of BounceNet’s ability to correctly estimate interference. We choose 100 different pairs of links from our testbed and measure the ground truth interference between every pair. For each pair, we consider both the direct path and indirect paths. To obtain the interference estimates from BounceNet, we perform the association phase using the experimental setup. Then, we use the measurements to find all the paths and compute the INR as described in section 5.2. Fig. 14(b) shows the CDF of the absolute error between the ground truth interference measurements and the

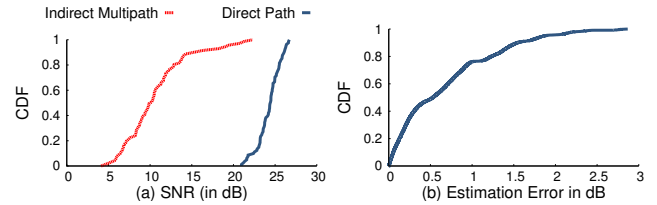


Figure 14: **Microbenchmarks:** (a) SNR of indirect vs. direct paths. (b) Interference estimation error.

Table 1: Percentage of Links with n Reflected Paths

| Room | $n = 0$ | $n = 1$ | $n = 2$ | $n = 3$ | $n = 4$ |
|--------------|---------|---------|---------|---------|---------|
| Lecture Hall | 0 | 20 | 46.6 | 26.6 | 6.6 |
| Atrium | 5 | 95 | 0 | 0 | 0 |
| Lounge | 0 | 46.6 | 50 | 3.3 | 0 |
| Empty Room | 0 | 21.0 | 52.6 | 26.4 | 0 |
| Lab | 0 | 37.4 | 41.4 | 21.2 | 0 |
| Office Space | 0 | 30 | 45 | 15 | 5 |

estimated values from BounceNet. BounceNet’s median error is 0.52 dB and 90th percentile error is 1.54 dB which is within the 3 dB tolerance for various mmWave MCSs. BounceNet is able to achieve such high accuracy in predicting the interference in the network because it accounts for both the multipath in the environment as well as the imperfections in antenna beam patterns. Furthermore, it is able to do this using only a linear number of measurements $O(N)$, therefore avoiding the need to explicitly measure interference between every pair which would be $O(N^2)$.

C. BounceNet’s Signal Routing

In Fig. 13, we present additional examples of BounceNet’s beam alignments in the 12° testbed. We pick one client configuration and plot the beam alignments computed by BounceNet for the first three time slots. We can see that BounceNet makes use of both direct and reflected paths in order to squeeze in as many links as possible for communication during the time slot. Furthermore, over the three time slots, BounceNet schedules the direct paths for different clients, thus clients get a chance to use their direct paths in different time slots. Clients that create less interference such as C1 and C10 get to use their direct paths in all time slots whereas clients that create more interference such as C2 or C7 get to use it once.

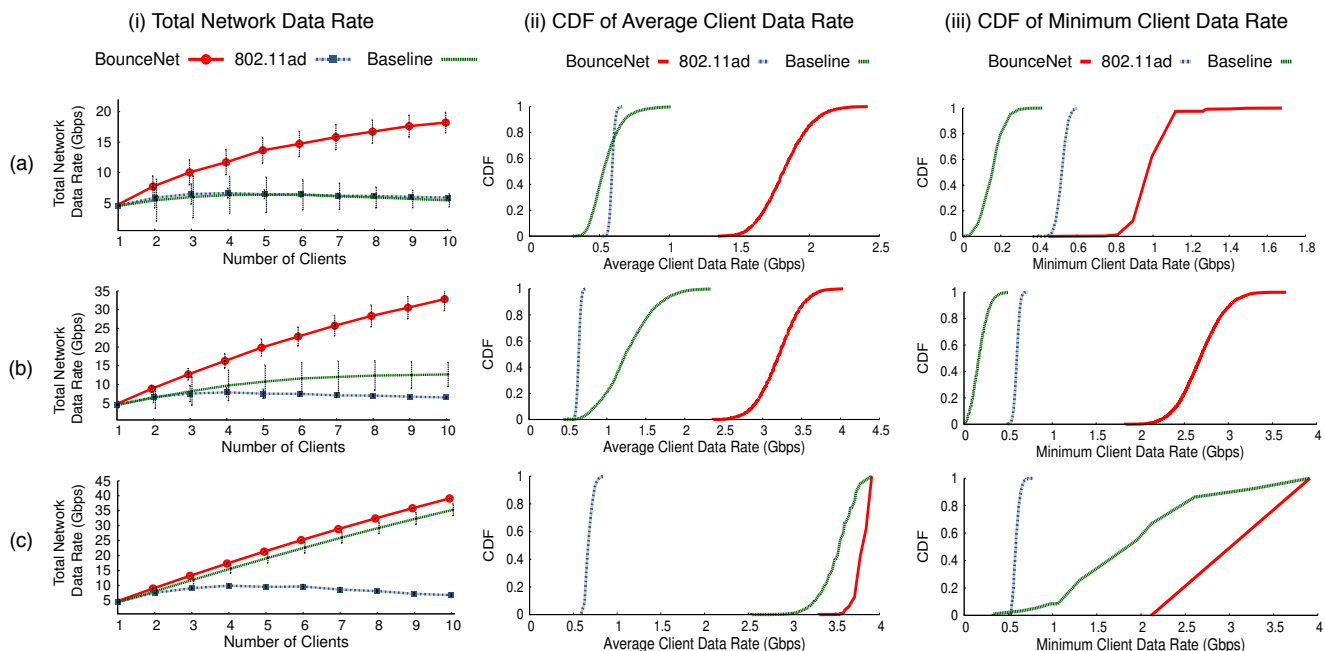


Figure 15: Data rates in BounceNet, 802.11ad and baseline for (a) 24 GHz phased array (b) 60 GHz with 12° beams (c) 60 GHz with 3°.

9 Evaluation Results

We will present our main evaluation results here. We will start by describing our baselines and evaluation metrics.

A. Compare Schemes: We compare BounceNet to:

- (1) **802.11ad with Spatial Reuse:** As described in section 3, the current standard provides a greedy mechanism for exploiting spatial reuse by measuring pairwise mutual interference and merging links that do not interfere into the same slots. If the nodes detect changes in the interference in the network, they reset to transmitting in exclusive time slots.
- (2) **Baseline:** Our baseline will consider independently aligning the beams of each AP and client and letting them transmit. To give the baseline an edge, we assume that the APs and clients can perform their beam search without creating any interference. Hence, they can find the right alignment in $O(N)$ and then use it for data transmission.

B. Metrics: We evaluate BounceNet using these metrics:

- **Total Network Data Rate:** The aggregate data rate of all the clients in the network.
- **Average Client Data Rate:** The average data rate of the clients in the network.
- **Minimum Client Data Rate:** The minimum data rate among all clients in the network.
- **Fraction of Time on the Channel:** The fraction of time slots a client gets to transmit in; used to evaluate fairness.
- **Average Client Throughput:** The average application layer throughput of a client using TCP or UDP flows.

C. BounceNet Data Rate Gain:

We start by evaluating the gains in total network data rates. Fig. 15(i) shows the total network data rate as a function of the number of clients in a network with 10 APs for BounceNet, 802.11ad, and the baseline. As the number of clients increases, BounceNet is able to scale the total network data rate with the number of clients to deliver a total of 39.2 Gbps and 32.8 Gbps data rates for 10 clients using 60 GHz with 3° and 12° beams respectively. For 24 GHz, BounceNet is able to achieve 18.2 Gbps for 10 clients. This is expected as sidelobe leakage of phased arrays creates more interference in the network which limits spatial reuse.

802.11ad, on the other hand, is unable to properly exploit spatial reuse and shows limited gains. Specifically, for the case of 10 clients, BounceNet achieves 6.6×, 5×, and 3.1× gain in network throughput as compared to 802.11ad for 3° beam, 12° beam, and the phased array respectively. This is due to 802.11ad’s inefficiency which stems from requiring pairs of links to measure mutual interference during data transmission and merge these links during the following beacon interval only if they do not interfere. The baseline can exploit spatial reuse for 3° beam since the interference in this case is very limited. Hence, for 10 clients with 3° beam, BounceNet only achieves 1.27× gain over the baseline. This gain, however, increases to 2.7× and 3.4× for 12° beam and the phased array respectively where there is more interference. In fact, the baseline is unable to exploit spatial reuse and scale network throughput in such cases.

In Fig. 15(ii) we plot the CDF of the average data rate achieved by the clients across all the runs with 10 clients in

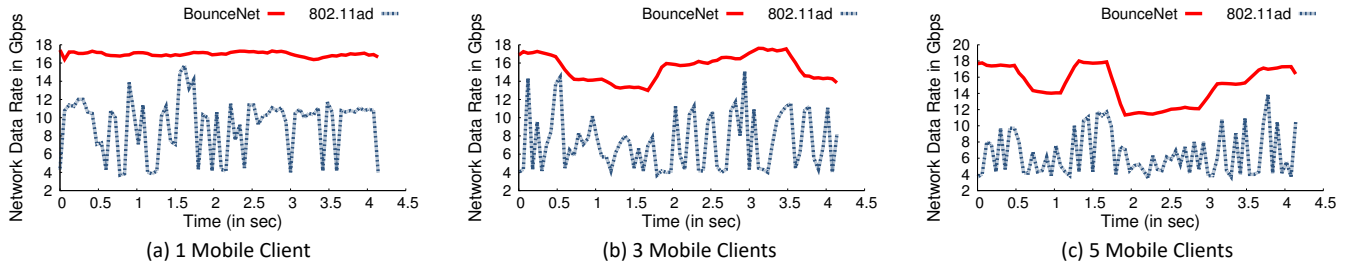


Figure 16: Mobility: This figure shows that BounceNet can adapt to changing and mobile clients whereas 802.11ad is unable to exploit spatial reuse in mobile networks.

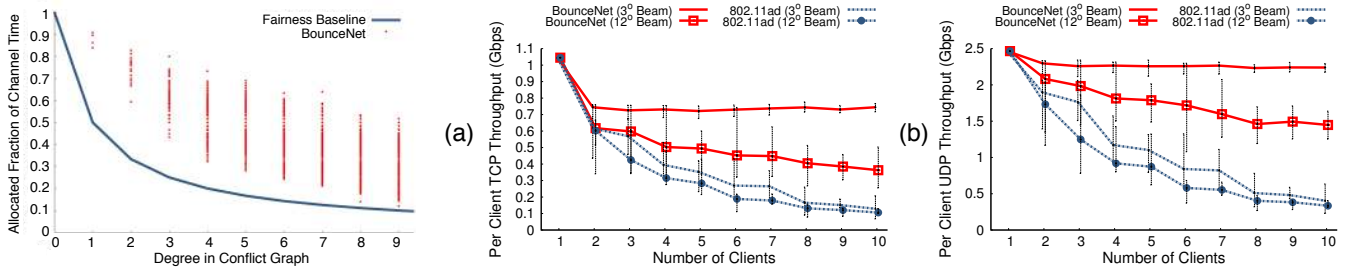


Figure 17: Client’s share of time on the channel.

Figure 18: BounceNet’s Application Level Average Throughput Under (a) TCP and (b) UDP.

the network. A client in BounceNet can achieve a 50th percentile average data rate of 3.8 Gbps for 3° beam, 3.25 Gbps for 12° beam, and 1.81 Gbps for the phased array. Whereas in 802.11ad, the 50th percentile average data rate is around 0.6 Gbps in all three cases. The baseline, however, shows high average data rate of 3.4 Gbps for 3° beam which decreases to 1.26 Gbps for 12° and 0.5 Gbps for the phased array. Hence, with wider beams, simply ignoring interference would result in an even worse performance than 802.11ad.

Two points are worth noting. First, each of the 10 clients in BounceNet can achieve a 90th percentile average data rate of 3.9 Gbps for 3°, 3.7 Gbps for 12°, and 2 Gbps for the phased array. This is a small deviation from the median data rate which shows that BounceNet is fair in dividing the rate across the clients. Second, while BounceNet scales the network throughput, the overhead of beam alignment starts to kick in. This, however, can be addressed by employing faster beam alignment protocols [23, 49, 64].

We also plot the CDF of the minimum data rate among all clients in Fig. 15(iii), across all the runs with 10 clients in the network. The figure shows that BounceNet can significantly improve the minimum and benefit worst case clients which can suffer from interference. BounceNet can improve the minimum data rate of any client in the network by 13.5× for 12° beam and 7.5× for phased arrays as compared to the baseline. This is because the baseline does not try to avoid interference, and hence clients that suffer from interference can really benefit from BounceNet.

In Appendix B, we present additional results when there are only 5 APs in the network. This allows us to evaluate

BounceNet in scenarios where clients outnumber the APs.

D. Adapting to Changes and Mobile Clients:

To understand BounceNet’s ability to adapt to mobile clients, we examine what happens to the total network data rate as clients move for both BounceNet and 802.11ad. As the baseline does not actively try to optimize for spatial reuse, we expect the total network data rate to remain smooth albeit lower than BounceNet.

We run an experiment where there are five clients in the network and we vary the number of clients that are moving. Fig. 16 shows the total network data rate versus time, when one client, three clients or five clients are moving. This figure shows that BounceNet can continue to maintain a high data rate as the clients move. For one client moving, BounceNet achieves almost a constant data rate. As more clients move, the interference patterns in the network change, and hence the maximum achievable data rate changes. The figure shows that BounceNet can quickly adapt to changes and continue to exploit spatial reuse.

On the other hand, the data rate in 802.11ad fluctuates significantly and keeps falling back to the case of no spatial reuse. This is because 802.11ad merges AP-client pairs only after measuring the mutual interference during the data transmission phase. Hence, it takes 802.11ad several beacon intervals (≈ 100 ms) to exploit spatial reuse. By that time, the client has moved and the interference patterns have changed. Even if one client moves, it can affect the interference patterns of many links. Fig. 16 shows that as more clients move, the interference patterns change faster, and hence 802.11ad is unable to properly exploit spatial reuse.

E. BounceNet Fairness:

Recall from section 6.2 that fairness in mmWave networks depends on how much each client interferes with other clients. If a client interferes with d other links, it should get at least a fraction of $1/(d+1)$ of time on the channel. For each of our 5000 experiments, we compute the fraction of channel time that a client interfering with d other links in the network obtains as a result of BounceNet's algorithm. Fig. 17 plots this fraction for all clients against their degree in the conflict graph (equivalent to their number of interfering links). The figure shows that the algorithm guarantees that all points lie above the line denoted by $Fraction = 1/(d+1)$. Hence, every link gets at least its fair share of channel time in BounceNet.

F. Application Level throughput in BounceNet:

In order to understand whether BounceNet's gains translate to higher layer network throughput, we evaluated the application level throughput achieved using BounceNet and 802.11ad under TCP and UDP traffic flows in ns3. Fig. 18 shows the throughput versus the number of clients. BounceNet's scaling properties are maintained with roughly the same gain over the 802.11ad standards. For 10 links, BounceNet can achieve a UDP throughput of 1.44 Gbps for 60 GHz with 12° beamwidth and 2.23 Gbps for 3° beamwidth. As expected, the application level throughput is lower than the MAC data rates due to the overhead of headers. For TCP the throughput is even lower with 360 Mbps for 12° beamwidth and 740 Mbps for 3° beamwidth. This is expected as TCP has larger overhead and does not perform well in wireless networks.

G. Results Summary:

802.11ad requires multiple beacon intervals to detect interference in the network and schedule concurrent transmissions. While this would work in completely static scenarios where the paths do not change, it is inefficient in mobile or dynamic environments. Our results show that in such cases, 802.11ad keeps resetting to a configuration with no spatial reuse. BounceNet, on the other hand, is able to maintain an up-to-date view of the paths and interference every Beacon Interval which allows it to achieve significant gains especially for narrower beams (e.g. 3°) where the potential for spatial reuse is very high.

The baseline, on the other hand, performs well with narrow beams (e.g. 3°) and on average achieves comparable results to BounceNet. However, the tail of the distribution is very long. Specifically, clients that experience interference would achieve significantly lower data rates than both BounceNet and 802.11ad. The performance quickly degrades for wider beams where there is more interference between links. BounceNet can achieve the best of both worlds by combining efficient path learning and interference estimation algorithms with signal routing and beam alignment. Hence, BounceNet can exploit spatial reuse for both very narrow beams and wide beams and can perform well in both static and mobile environments.

10 Limitations and Discussion

Few points are worth noting.

- Our current evaluation is limited by today's hardware which makes it infeasible to implement a full-fledged real-time version of our system. Cheap commercial mmWave devices [2, 39, 56] do not provide access to the lower layers: PHY and MAC. On the other hand, the hardware we used costs around \$14,000 for the RF front end of one TX/RX pair, making it prohibitively expensive to scale the implementation. Note, however, that our simulations are not based on ray-tracing or any channel modeling. Rather, they are based on actual measurements of SNRs and beam scanning through a labor-intensive study that generated over 5000 configurations. We have also used two pairs of links to verify that our interference estimates are accurate. Our results show a significant opportunity to scale the throughput in mmWave networks and we believe the protocol can be implemented on cheap commercial devices if the chip manufacturers open up the firmware.
- BounceNet's protocol is mainly designed for continuous traffic in applications like VR, 3D video streaming, and Robotics. To deal with bursty traffic, one can leverage the polling mechanism available in 802.11ad [26] to obtain a *real-time* view of the traffic demands for different clients during the Beacon Interval, and adjust the conflict graph based on the traffic.
- BounceNet's interference estimation relies on accurate measurements of the SNR. The high directionality in mmWave networks reduces multipath fading and channel fluctuations which allows us to achieve accurate estimates as we show in section 8. However, to address the case of noisy and unstable SNR measurements, we take a more conservative approach for determining when two links interfere (Section 6.2.A). The threshold to determine interference can be adjusted as a trade-off between robustness to noisy SNR estimates and maximizing spatial reuse.

11 Conclusion

In this paper, we introduced BounceNet, the first many-to-many millimeter wave beam alignment system that can efficiently align the beams of many APs and clients in a manner that allows them to simultaneously communicate without interfering. We evaluated BounceNet using three experimental testbeds and demonstrated that it can enable dense spatial reuse and scale the total network throughput with the number of APs and clients.

Acknowledgments: We would like to thank our shepherd, Prof. Lin Zhong, and the reviewers for their comments. We would also like to thank Piotr Indyk for his input on the proof. Lastly, we also thank the Systems and Networking Group (SyNRG) at UIUC for their feedback. This work is funded in part by NSF Award CNS-1750725.

References

- [1] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Enabling high-quality untethered virtual reality. In *NSDI*, 2017.
- [2] Acer. TravelMate P6 TMP658-M-70S3 Laptop.
- [3] Xueli An and Ramin Hekmat. Directional MAC protocol for millimeter wave based wireless personal area networks. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 1636–1640. IEEE, 2008.
- [4] Xueli An, Shuang Zhang, and Ramin Hekmat. Enhanced mac layer protocol for millimeter wave based WPAN. In *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5. IEEE, 2008.
- [5] Christopher R. Anderson and Theodore S. Rappaport. In-Building Wideband Partition Loss Measurements at 2.5 and 60 GHz. *IEEE TWC*, 2004.
- [6] Hany Assasa and Joerg Widmer. Extending the IEEE 802.11ad model: Scheduled access, spatial reuse, clustering, and relaying. In *Proceedings of the Workshop on Ns-3, WNS3 '17*, pages 39–46, New York, NY, USA, 2017. ACM.
- [7] Matt Branda. Qualcomm Research demonstrates robust mmWave design for 5G. Qualcomm Technologies Inc., November 2015.
- [8] Neeraj Choubey and Ali Yazdan Panah. Introducing Facebook’s new terrestrial connectivity systems-Terragraph and Project ARIES, Facebook Research, 2016.
- [9] Romit Roy Choudhury and Nitin H Vaidya. Deafness: A MAC problem in ad hoc networks when using directional antennas. In *Proceedings of the 12th IEEE International Conference on Network Protocols.*, pages 283–292, 2004.
- [10] Romit Roy Choudhury, Xue Yang, Ram Ramanathan, and Nitin H Vaidya. On designing MAC protocols for wireless networks using directional antennas. *IEEE Transactions on Mobile Computing*, 5(5):477–491, 2006.
- [11] Adam Connor-Simons. Enabling wireless virtual reality, MIT News, 2016.
- [12] Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Chao Zhu, Xiangyang Li, Liu Yang, and Ning Ge. Diamond: Nesting the Data Center Network with Wireless Rings in 3D Space. In *NSDI*, 2016.
- [13] András Frank. Some polynomial algorithms for certain graphs and hypergraphs. In *Proceedings of the Fifth British Combinatorial Conference*, pages 211–226, 1975.
- [14] Y. Ghasempour, M. K. Haider, and E. W. Knightly. Decoupling beam steering and user selection for MU-MIMO 60-GHz WLANs. *IEEE/ACM Transactions on Networking*, pages 1–14, 2018.
- [15] Yasaman Ghasempour, Claudio RCM da Silva, Carlos Cordeiro, and Edward W Knightly. IEEE 802.11ay: Next-generation 60 GHz communication for 100 Gb/s Wi-Fi. *IEEE Communications Magazine*, 55(12):186–192, 2017.
- [16] Yasaman Ghasempour, Muhammad K Haider, Carlos Cordeiro, Dimitrios Koutsonikolas, and Edward W Knightly. Multi-stream beam-training for mmwave mimo networks. In *ACM MobiCom*, 2018.
- [17] Michelle X Gong, Dmitry Akhmetov, Roy Want, and Shiwen Mao. Directional CSMA/CA protocol with spatial reuse for mmWave wireless networks. In *Global Telecommunications Conference (GLOBECOM)*, pages 1–5. IEEE, 2010.
- [18] Michelle X Gong, Robert Stacey, Dmitry Akhmetov, and Shiwen Mao. A directional CSMA/CA protocol for mmWave wireless PANs. In *Wireless Communication and Networking Conference*, pages 1–6. IEEE, 2010.
- [19] Muhammad Kumail Haider, Yasaman Ghasempour, and Edward W Knightly. Search light: Tracking device mobility using indoor luminaries to adapt 60 GHz beams. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 181–190. ACM, 2018.
- [20] Muhammad Kumail Haider, Yasaman Ghasempour, Dimitrios Koutsonikolas, and Edward W Knightly. Lister: mmwave beam acquisition and steering by tracking indicator leds on wireless aps. 2018.
- [21] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. Augmenting Data Center Networks with Multi-Gigabit Wireless Links. In *ACM SIGCOMM*, 2011.
- [22] Ezzeldin Hamed, Hariharan Rahul, Mohammed A Abdelghany, and Dina Katabi. Real-time Distributed MIMO Systems. In *Proceedings of ACM SIGCOMM, 2016*, pages 412–425.
- [23] Haitham Hassanieh, Omid Abari, Michael Rodriguez, Mohammed Abdelghany, Dina Katabi, and Piotr Indyk. Fast millimeter wave beam alignment. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 432–445. ACM, 2018.
- [24] Kelly Hill. A look at Verizon’s fixed millimeter wave testing. RCR Wireless News, May 2017. <http://www.rcrwireless.com/20170501/test-and-measurement/verizon-fixed-millimeter-wave-testing-tag6>.
- [25] Huawei. Huawei to bring 73GHz mmWave Mu-MIMO live demo to Deutsche Telekom, Press Release, 2016.
- [26] IEEE Standards Association. IEEE Standards 802.11ad-2012: Enhancements for Very High Throughput in the 60 GHz Band, 2012.
- [27] Intel Inc. Intel accelerates path to 5G, Press Release, 2016.
- [28] Gentian Jakllari, Wenjie Luo, and Srikanth V Krishnamurthy. An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas. *IEEE Transactions on Wireless Communications*, 6(3):1114–1024, 2007.
- [29] Suraj Jog, Jiaming Wang, Haitham Hassanieh, and Romit Roy Choudhury. Enabling Dense Spatial Reuse in mmWave Networks. In *ACM SIGCOMM*, 2018.
- [30] Sanjeev Khanna and Krishnan Kumaran. On wireless spectrum estimation and generalized graph coloring. In *Proceedings of INFOCOM*, volume 3, pages 1273–1283. IEEE, 1998.
- [31] Sungoh Kwon and Joerg Widmer. Relay selection for mmwave communications. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, pages 1–6. IEEE, 2017.
- [32] Sungoh Kwon and Joerg Widmer. Multi-beam power allocation for mmwave communications under random blockage. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2018.
- [33] Xi Liu, Anmol Sheth, Michael Kaminsky, Konstantina Papagiannaki, Srinivasan Seshan, and Peter Steenkiste. DIRC: Increasing indoor wireless capacity using directional antennas. *ACM SIGCOMM Computer Communication Review*, 39(4):171–182, 2009.
- [34] Barry Manz. 5G Cellular Networks Are the Future of Robotics, Mouser Electronics, 2016.
- [35] Markets and Markets. Millimeter Wave Technology Market worth 4,632.8 Million USD by 2022, Press Release, 2017.
- [36] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for WLANs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):19–31, 2005.
- [37] Thomas Moon, Junfeng Guan, and Haitham Hassanieh. Online millimeter wave phased array calibration based on channel state information. In *IEEE VLSI Test Symposium, 2019. VTS'19*. IEEE, 2019.
- [38] Raghuraman Mudumbai, SK Singh, and Upamanyu Madhow. Medium access control for 60 GHz outdoor mesh networks with highly directional links. In *INFOCOM 2009, IEEE*, pages 2871–2875. IEEE, 2009.
- [39] NetGear. NIGHTHAWK X10 R9000 Wi-Fi Router.
- [40] Thomas Nitsche, Carlos Cordeiro, Adriana B Flores, Edward W Knightly, Eldad Perahia, and Joerg C Widmer. IEEE 802.11 ad: directional 60 GHz communication for multi-gigabit-per-second Wi-Fi. *IEEE Communications Magazine*, 52(12):132–141, 2014.

- [41] Thomas Nitsche, Adriana B Flores, Edward W Knightly, and Joerg Widmer. Steering with eyes closed: mm-wave beam steering without in-band measurement. In *Conference on Computer Communications (INFOCOM)*, pages 2416–2424. IEEE, 2015.
- [42] Joan Palacios, Guillermo Bielsa, Paolo Casari, and Joerg Widmer. Communication-driven localization and mapping for millimeter wave networks. *IEEE INFOCOM*, 2018.
- [43] Pasternack Enterprises Inc. 60 GHz Transmitter/Receiver Development System. www.pasternack.com.
- [44] Jian Qiao, Lin X Cai, Xuemin Shen, and Jon W Mark. STDMA-based scheduling algorithm for concurrent transmissions in directional millimeter wave networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 5221–5225. IEEE, 2012.
- [45] Qualcomm. QCA9500 specifications. <https://www.qualcomm.com/products/qca9500>.
- [46] Hariharan Shankar Rahul, Swarun Kumar, and Dina Katabi. Jmb: scaling wireless capacity with user demands. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 235–246. ACM, 2012.
- [47] Krishna N Ramachandran, Elizabeth M Belding-Royer, Kevin C Almeroth, and Milind M Buddhikot. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In *Infocom*, volume 6, pages 1–12, 2006.
- [48] Sundeep Rangan, Theodore S Rappaport, and Elza Erkip. Millimeter-wave cellular wireless networks: Potentials and challenges. *IEEE*, 2014.
- [49] Maryam Eslami Rasekh, Zhinus Marzi, Yanzi Zhu, Upamanyu Madhow, and Haitao Zheng. Noncoherent mmwave path tracking. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, pages 13–18. ACM, 2017.
- [50] Wonil Roh, Ji-Yun Seol, Jeongho Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and Farshid Aryanfar. Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results. *IEEE Communications Magazine*, 52(2):106–113, 2014.
- [51] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. Centaur: realizing the full potential of centralized wlns through a hybrid data path. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 297–308. ACM, 2009.
- [52] Sumit Singh, Federico Ziliotto, Upamanyu Madhow, E Belding, and Mark Rodwell. Blockage and directivity in 60 GHz wireless personal area networks: From cross-layer model to multihop MAC design. *IEEE Journal on Selected Areas in Communications*, 27(8):1400–1413, 2009.
- [53] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parameswaran Ramanathan. 60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling. In *SIGMETRICS*, 2015.
- [54] Sanjib Sur, Xinyu Zhang, Parameswaran Ramanathan, and Ranveer Chandra. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *NSDI*, 2016.
- [55] Tensorcom. Product specifications. <http://tensorcom.com/products-1/>.
- [56] TP-Link. Talon AD7200 Multi-Band Wi-Fi Router.
- [57] Victoria Turk. These Supermarket Warehouse Robots Have Their Own Mobile Network, Vice Motherboard, 2016.
- [58] Kiran Venugopal, Matthew C Valenti, and Robert W Heath. Interference in finite-sized highly dense millimeter wave networks. In *Information Theory and Applications Workshop (ITA), 2015*, pages 175–180. IEEE, 2015.
- [59] Weizhao Wang, Yu Wang, Xiang-Yang Li, Wen-Zhan Song, and Ophir Frieder. Efficient interference-aware TDMA link scheduling for static wireless networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 262–273. ACM, 2006.
- [60] Teng Wei and Xinyu Zhang. Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support. In *MobiCom'17*, 2017.
- [61] Teng Wei, Anfu Zhou, and Xinyu Zhang. Facilitating Robust 60 GHz Network Deployment By Sensing Ambient Reflectors. In *NSDI*, 2017.
- [62] Ding Zhang, Mihir Garude, and Parth H Pathak. mmChoir: Exploiting joint transmissions for reliable 60 GHz mmwave WLANs. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 251–260. ACM, 2018.
- [63] Anfu Zhou, Leilei Wu, Shaoqing Xu, Huadong Ma, Teng Wei, and Xinyu Zhang. Following the shadow: Agile 3-d beam-steering for 60 GHz wireless networks. *IEEE INFOCOM*, 2018.
- [64] Anfu Zhou, Xinyu Zhang, and Huadong Ma. Beam-forecast: Facilitating Mobile 60 GHz Networks via Model-driven Beam Steering. In *INFOCOM*, 2017.
- [65] Anfu Zhou, Xinyu Zhang, and Huadong Ma. Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.
- [66] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H Zheng. Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *ACM SIGCOMM*, 2012.

Appendix

A Proof of Lemma 6.1

Suppose we are given a graph $G(V, E)$ where $|V| = N$ and $d(u)$ denotes the degree of u . Consider the following process which iteratively assigns weights (in the range $\{0 \dots M\}$) to the vertices. The initial assignment is F_0 such that $F_0(v) = M$ for all $v \in V$. We compute F_t as follows:

- Compute a Weighted Max Independent Set W_{t+1} in the weighted graph induced by G and F_t .
- If $u \in W_{t+1}$, then $F_{t+1}(u) = F_t(u) - (d(u) + 1)$ if $F_t(u) > 2(d(u) + 1)$ and $F_{t+1}(u) = 0$ otherwise.
- If $u \notin W_{t+1}$, then $F_{t+1}(u) = F_t(u)$.

Lemma A.1 *If $t = O(M \log(NM))$, then $F_t(u) = 0 \forall u \in V$*

Proof Consider the potential function $T_t = \sum_u F_t(u)$.

Claim A.2 $T_{t+1} \leq T_t(1 - 1/M)$.

Proof Consider the set of vertices S_t containing u 's such that $F_t(u) > 0$. Since the maximum value of $F_t(u)$ is M , it follows that

$$|S_t| \geq T_t/M \quad (2)$$

Consider now the set W_{t+1} , and w.l.o.g. assume that $W_{t+1} \subset S_t$. Observe that W_{t+1} must be a *maximal* independent set, i.e., we cannot add any $u \in S_t - W_{t+1}$ to W_{t+1} without violating the independence property. Since the total number of nodes

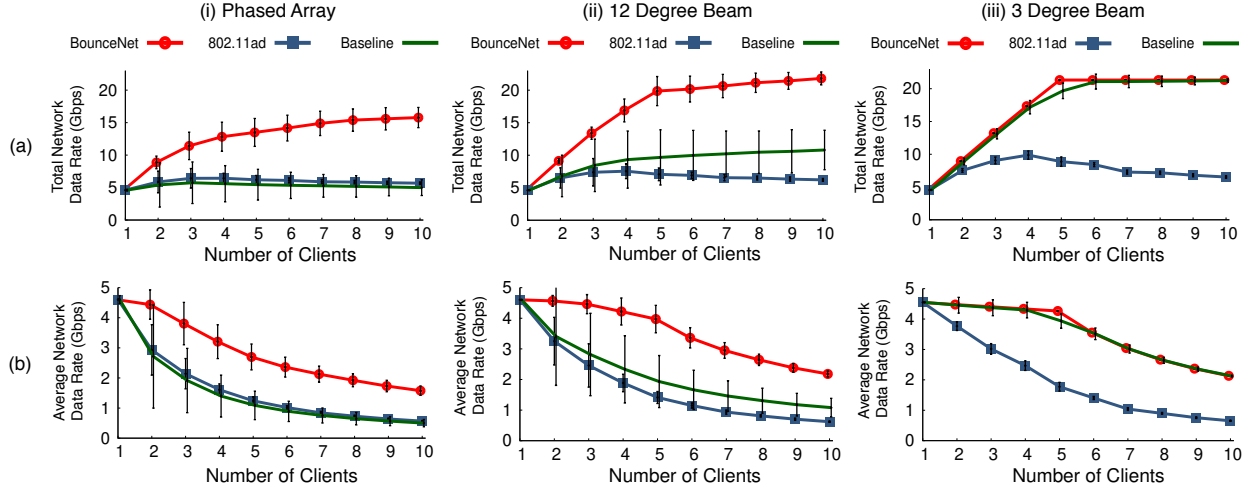


Figure 19: Data rates in BounceNet, 802.11ad and baseline for the case of 5 APs in network (a) Total Network Data Rates (b) Average Client Data Rates.

with an edge to a node in W_{t+1} (including self-loops) is at most $\sum_{w \in W_{t+1}} d(w) + 1$, it follows that

$$\sum_{w \in W_{t+1}} d(w) + 1 \geq |S_t| \quad (3)$$

However, the left-hand side in the above expression is upper bounded by the amount by which we reduce the potential, i.e., by the difference $T_t - T_{t+1}$ (the reduction in potential could be higher, because we round all weights smaller than $d + 1$ to 0). From Equations 2 and 3 we have

$$T_t - T_{t+1} \geq \sum_{w \in W_{t+1}} d(w) + 1 \geq |S_t| \geq T_t/M$$

and the lemma follows.

Since T_t has integral values, it follows that after $O(M \log(T_0))$ steps we have $T_t = 0$, and therefore $F_t(u) = 0$ for all u .

B Data Rate Gains for 5 APs

In Fig. 19, we present results for the case when there are 5 APs in the network. This allows us to evaluate BounceNet’s performance in scenarios where the number of clients is greater than the number of APs. In such scenarios where the clients outnumber the APs, two or more clients could be assigned to the same AP, following the algorithm presented in Section 6.1. Since clients that share an AP essentially be considered as interfering links, the corresponding nodes in the conflict graph will have edges between them. We can then apply BounceNet’s signal routing algorithm (Section 6.2 and 6.3) to this modified conflict graph.

Fig. 19(a) shows the total network data rate, and Fig. 19(b) shows the average network data rate per client, as a function

of the number of clients in the network. BounceNet is able to deliver a total of 21.33 Gbps, 20.81 Gbps and 15.78 Gbps data rates for 10 clients in the 3° beam, 12° beam and the phased array testbeds respectively. The baseline performs almost as well as BounceNet for the 3° beam since the interference in this case is very limited, and as a result, the baseline is able to exploit spatial reuse. However, as the amount of interference increases, the performance of the baseline deteriorates, with BounceNet achieving $2.2\times$ and $3.2\times$ gain in network throughput over the baseline for the case of 10 clients in the 12° beam, and the phased array testbeds respectively. Since the baseline does not account for interfering links, it leads to frequent packet collisions, and as a result, inefficient use of the channel.

Compared to 802.11ad, BounceNet achieves $3.26\times$, $3.35\times$, and $2.78\times$ gain in network throughput for the case of 10 clients in the 3° beam, 12° beam, and the phased array testbed respectively. One should note that for 802.11ad, the gains with 5 APs are smaller as compared to the gains observed in Section 9.C, where there were 10 APs in the network. This is because BounceNet’s strength over 802.11ad comes primarily from its ability to exploit spatial reuse efficiently, and with only 5 APs in the network, the potential for spatial reuse is reduced, and therefore the gains that BounceNet can provide over the standard will be smaller. Hence, to achieve significant gains in throughput, BounceNet advocates for dense AP deployments with narrow directional antenna beams in mmWave networks.

Finally, the following points are worth noting.

- With the 3° beam in the 60 GHz testbed, we see that the total network data rate for BounceNet saturates after 5 clients as can be seen in Fig. 19(a)(iii), achieving 21.33 Gbps and 21.29 Gbps for 10 clients and 5 clients respectively. This

is expected, since at any given time at most 5 clients can be communicating simultaneously in the network. Such saturation can also be observed in the other two testbeds.

- It may seem counter-intuitive that the total network data rate for BounceNet in the 12° and the phased array testbeds continues to grow even when there are more than 5 clients in the network. This happens because as the number of clients increases in the network, the total number of propagation paths (direct and reflected) between APs and clients

increases as well. Therefore, now it is more likely that BounceNet can find a set of five propagation paths that can coexist in the network, and consequently, BounceNet can schedule more clients in every time slot. However, one should note that the rate of growth of the network data rate reduces as the number of clients increases beyond five, and correspondingly, the average per-client data rates start to drop more sharply beyond five clients as can be seen in Fig. 19(b).