Making the Right Connections: Multi-AP Association and Flow Control in 60GHz Band

Fan Zhou, M. Yousof Naderi, Kunal Sankhe, and Kaushik Chowdhury
Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA
Email: zhou.fan1@husky.neu.edu, naderi@coe.neu.edu, sankhe.ku@husky.neu.edu, krc@ece.neu.edu

***Abstract*—Emerging network architectures in 60GHz millimeter wave bands will likely use dense deployment of access points (APs) given the high attenuation and frequent line-of-sight related outages. We show in this paper that naively connecting to any available AP, or even multiple APs, may not fully realize the promise of efficiently utilizing the extremely large bandwidths available in this band. This paper holistically addresses these problems through the Multi-AP Association Protocol (MAP) that: (i) indicates ideal durations for beam-searching at the physical/link layers of the protocol stack that will result in minimal interruptions to user-traffic, (ii) devises a Multi-AP association framework based on multipath TCP, which increases the network robustness by allowing immediate redirection of traffic to alternative APs whenever an existing connection is interrupted, and (iii) designs an exploration-exploitation aware flow scheduling algorithm that dynamically activates the subflows to optimize transport layer performance. MAP is a lightweight, client side system that needs zero modifications to existing protocol stack, though it interacts with the latter to opportunistically trigger standards-defined functions. The paper also presents convergence analysis of throughput, experimental validation on a 60GHz network testbed, and trace-driven simulation studies that shows MAP achieving 5-7x reduction in re-buffering rate in HD video streaming, rapid convergence to the best possible AP, and fair allocation of resources among clients.**

I. Introduction

Network densification and vast spectrum availability in the 60GHz band is seen as a promising solution to meet the surging demands of next generation short-range wireless networks. Indeed, the benefits of using these bands over the 2.4/5GHz bands range from reduced interference to over-2GHz single channel bandwidth [1]. However, there are also associated challenges introduced by the high rates of signal attenuation in the millimeter wave range, reported to be 5-144x times that of the 2.4/5GHz bands [2]. This limits associations based on line of sight (LoS) between a given transceiver pair and requires concentrating the transmission energy of the antenna via beamforming to increase the received signal power. We argue in this paper that solely focusing on the physical/link layer is not sufficient to efficiently utilize 60GHz connections.

• **Limitations of existing state of the art:** Today's 802.11ad standard incorporates a beam-searching mechanism that is triggered *after* the current beam is blocked. This is a time-consuming process, as it involves an exhaustive antenna sector sweeping over all spaces for the optimal beam direction, which can take up to a few seconds and significantly degrades the user's quality of experience (QoE) [3]. Several innovative physical/link layer approaches have been proposed to shorten the beam-searching time ([4]–[6]). However, we believe that considering additional inputs from the application, transport and network layers to guide lower layer actions can yield significant benefits to standards such as 802.11ad and beyond (quantitatively studied in Sec.II).

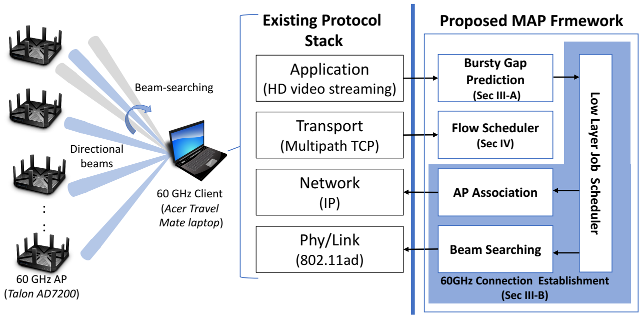


Fig. 1: Interconnections between MAP and regular protocol stack

1) Static AP-client associations: After the client connects to a given 802.11ad AP, it does not explore other, possibly better association options. Poor link conditions may reduce the effective link capacity to a fraction of its regular value over time, while the client remains connected.

2) Limited inter-layer feedback: LoS outages can be frequent [7], and the beam-searching period to re-establish a new link is considerably long. This also forces repeated attempts to establish new end-to-end TCP connections. This further deteriorates the user-QoE by adding to the network setup delay and under-utilizes the available bandwidth.

• **Proposed approach and innovation:** Our approach enables cross-layer connection management through the Multi-AP Association Protocol (MAP). It runs only at the client-side and does not need centralized support from the APs, thus avoiding overheads of added control signaling. Different from classical cross-layer approaches, MAP is implemented parallel to the regular network protocol stack (Fig. 1). MAP has access to layer-specific variables from the application and transport layers, and is also able to call out standards-supported functions at the network and Phy/Link layers. Thus, it optimizes the functioning of the existing protocol stack, without any software changes to it. As long as external function calls are supported, MAP can be implemented as a software module to an existing client. MAP only requires that Multipath TCP (MPTCP), IP and 802.11ad are choosen at the transport, network and Phy/Link layers, respectively, within the regular network stack.

• **MAP design overview:** MAP is composed of three parts, as shown in Fig. 1: (i) it analyzes application layer traffic patterns to predict bursty gaps in channel utilization, (ii) it proactively triggers beam-searching and AP association functions to identify the best links, using the channel availability gaps identified in (i), (iii) it optimizes the downlink throughput by dynamically scheduling sub-flows to appropriate paths via MPTCP and by maintaining pre-configured TCP connections that can be activated immediately upon sudden link failure. The above steps are explained in more detail as follows:

First, MAP leverages the inherently *bursty* traffic in today's mobile applications to identify when beam-searching functions should be conducted. It uses statistical measure of the packet delivery times at the client nodes to predict when gaps with no traffic may occur with high probability. MAP indicates these upcoming durations to the physical/link layer for scheduling additional AP association or beam-searching to avoid disrupting user-traffic. Thus, MAP maintains a vector of associated APs with freshly-trained beams, without impacting user QoE.

Second, MAP schedules sub-flows to one of many available paths (indicated by the vector of associated APs) to optimize the end-to-end performance. We recall that classical MPTCP opens sub-flows on all available paths to achieve a net aggregate rate that is the summation of the downlink rates of each AP. However, commercially available network products only use a single 60GHz Network Interface Card (NIC). Thus, each client can only transmit packets to one AP at a time. While MAP retains MPTCP's ability to setup connections through all available paths, it only activates one path that has the best observed throughput with other connections in a back-up mode. MAP formulates the path selection as an instance of the multi-armed bandit problem. It tunes the exploration-exploitation process to allow each client to quickly converge to the optimal AP despite unpredictable 60GHz link performance.

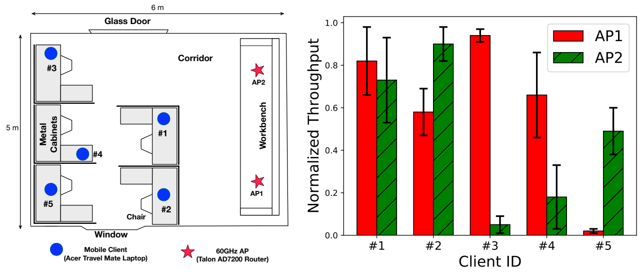
Summary of Contributions:

1) Based on insights from experimental studies, we propose MAP, which to the best of our knowledge, is the first cross-layer approach aimed at increasing the robustness of 60GHz networks by jointly exploiting the application traffic pattern and multi-AP association. MAP does not modify the protocol stack layers, but interacts with it to translate the benefits of densification to the end user.

2) We devise a method for estimating bursty mobile traffic at the application layer, and validate it using HD video streaming applications. We use these gaps to show how a list of optimized links with nearby APs can be pro-actively maintained.

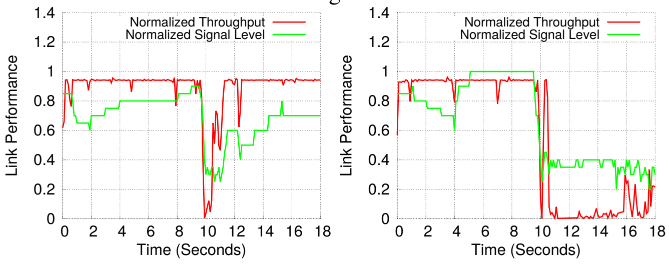
3) We design a lightweight, distributed flow selection algorithm using a multiarm bandit algorithm that achieves fast convergence with a derived upper confidence bound. We theoretically analyze MAP and prove that it can achieve performance that is near-optimal without global knowledge.

4) We implement and validate MAP through a trace-driven emulation on a 60GHz Linux testbed and perform additional city-scale validation studies using system-level simulation.



(a) (b)

Fig. 2: (a) Experimental setup in the lab; (b) Performance diversity observed in five clients connecting to two different APs



(a) (b)

Fig. 3: (a) Link performance recovers 2s after blockage; (b) Link fails to recover. The red and green lines show the normalized transport throughput (0-1Gbps) and PHY layer signal level (0-100%). The throughput is limited by the line rate of the NIC.

II. Preliminary Experiments and Motivation

In this section, we experimentally show the limitations of current 802.11ad based 60GHz links under a single-AP architecture, and discuss why a cross-layer design can help realizing the benefits of network densification. Then, we show how bursty traffic patterns of mobile applications can be exploited to build a robust 60GHz network.

A. Experimental setup

We use a Acer Travel Mate laptops as mobile clients and two Talon AD7200 routers as the 60GHz APs (see Fig. 2(a)). These devices use Qualcomm's tri-band network adapter, which includes a Wilocity 60GHz chipset and a 2*8 phased array antenna for beamforming. The AP is able to adjust beam-width and directions dynamically according to its built-in 802.11ad beam-forming mechanism. The PHY rate of the Wilocity chipset varies widely, and we have logged an observed range of [27.5Mbps,3.85Gbps], depending on signal quality. The experiments were performed in our lab, which represents a typical and challenging indoor environment for 60GHz connections with various sources of LoS blockage. In addition, there is a standard 2.4GHz 802.11b/g/n-based AP that serves as an *always available* link, and any of the clients can associate with it without beamforming.

B. User-QOE impact in single AP-managed 60GHz networks

•**Unreliable link recovery:** In this experiment, a mobile client is pre-configured to associate with a 60GHz AP, before we initiate a TCP flow from the local server that has a wired connection to the AP. Fig.3(a) shows the situation where the

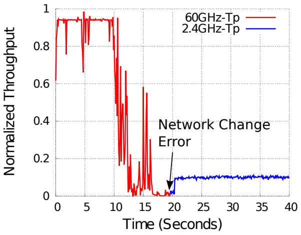


Fig. 4: Poor utilization of multi-APs leads to significant throughput loss.

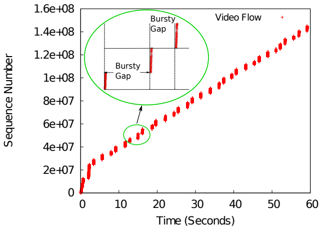


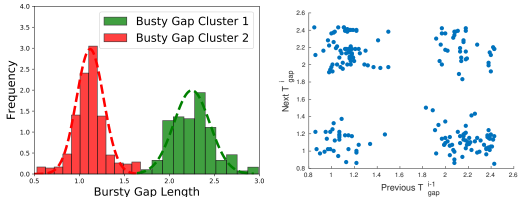
Fig. 5: Bursty traffic pattern of video streaming application

link is interrupted at the 10s mark.The time between the sharp fall in the (i) physical layer signal strength and the (ii) transport layer TCP throughput, and their subsequent recovery when the beam-searching procedure completes is about 2 seconds, a considerably long break for modern mobile applications.

Fig.3(b) depicts the situation where the signal level and TCP throughput never recover. This occurs with approximately 20% probability, indicating that the AP fails to find an alternate stable LoS path, or the beam-searching function is not triggered at all as the link can still sustain the lowest MCS rate. ***Implication:*** While beam-searching is necessary to maintain high-quality connection, this is a high-overhead operation and should ideally be completed before the application performance starts to drop. It also suggests that *a single 60GHz link may not always ensure LoS in all conditions.*

• **Poor utilization of multi-connections opportunity:** We start a TCP flow between the mobile client and server using an active 60GHz link. The client also has a secondary link to a traditional 2.4GHz AP. We then block the 60GHz link between the 10-30s mark, and plot the real time throughput in Fig.4. Though the client switches to the 2.4GHz WiFi network with 9x lower bandwidth, it does not automatically revert back after the blockage is removed (at the 30s mark). Secondly, we find that a *network change error* is triggered when the client switches to the 2.4GHz AP, which forces a restart in the TCP connection. Such an error is caused by the change in the client's IP address, as the destination IP is one of the five elements within the tuple that define a unique TCP socket. It can force the user to re-buffer the video-stream, or restart the FTP file transmission from the beginning. In either case, there is unnecessary bandwidth wastage. ***Implication:*** Without meaningful interaction between protocol layers, network densification may not produce performance benefits, thus motivating the cross-layer approach in MAP.

• **Performance variations over different 60GHz APs:** Since the link at 60GHz band is extremely sensitive to the channel conditions, clients may experience high variation in performance when connected to different APs. We measure the average downlink throughput of the clients at five different locations within a small cubical (5x6m) when connecting to two APs, shown in Fig. 2(a). We find in Fig. 2(b) that the client at location 3 can get optimal performance with AP1. However, its throughput drops to 0 at location 5 if it remains connected to AP1, despite spatial separation between these two locations being only 2m. ***Implication:*** Minor relative displacements can degrade the 60GHz link performance. Thus, the lower layers must continuously seek the optimal AP in the 60GHz band, and call upon the beam-searching process with greater frequency, albeit at the risk of disruption to the traffic transmission and QoE of the upper layers.



(a) (b)

Fig. 6: (a): Gaussian Mixture Model for the bursty gap length distribution; (b): Markov transition between bursty clusters (or *states*).

C. Leveraging transmission gaps: video streaming case study

As an example of a bandwidth intensive application, HTTP-based video streaming accounts for over 50% of the peak downlink traffic in the US and will likely be the key beneficiary of high bandwidth 60GHz links. Video traffic is separated into *chunks* and the typical length of a given chunk is 2-10s. The client requests each chunk *independently* and on-demand from the video server [8], which introduces random time intervals between two successive requests (see Fig.5). ***Implication:*** These bursty traffic patterns provides an opportunity to the lower layers to leverage available channel access opportunities.

III. Deadline-aware multi-AP Association

In this section, we show how MAP actively estimates the idle times within bursty traffic (Sec.III-A). This estimated duration serves as a *deadline* to complete a connection to a new AP and/or improve an existing connection by calling beam-searching functions for known APs (Sec.III-B).

A. Model-driven Bursty Gap Prediction

To statistically analyze the bursty gaps, we collect packet traces from 40+ hours HD video from one of the top video stream providers in the US with `tcpdump`. The laptop operates without cross-traffic, and is connected to the 60GHz AP with stable LoS connection (throughput > 600 Mbps).

• *Observation 1: The bursty gaps are distributed around multiple discrete values, which can be effectively captured with a Gaussian Mixture model.* As an example, Fig.6(a) plots the histogram of bursty gaps collected from one video session. The samples aggregate around 1s and 2.3s, which is approximated accurately with two Gaussian distributions. This motivates our goal to model the bursty gaps with several discrete *states*. Each state represents a different level of the burst gap length, which can be captured with a normal distribution $\mathcal{N}\ (\mu, \sigma^2)$. Since these states have discontinuous mean values, predicting the next bursty gap using either the historical average mean or the last sample is not accurate.

• *Observation 2: The transition of bursty gap states shows persistent properties of Markov process.* To predict bursty gaps, we need to know not only the distribution of different bursty gap states, but also the transitions between them. To do so, we study the length variation of two consecutive bursty gaps through a scatter plot represented by the two axes: the T_{gap}^{i-1} and T_{gap}^{i} ($(i-1)'th$ and $i'th$ bursty gap). Fig.6(b) provides a visual example of the trend where the states evolve in the form of a Markov chain. From the transition diagram, we find the bursty gap will move from one state (e.g, $\mu = 1s$) to a different one with probability $P = 2/3$ (e.g, $\mu = 2.3s$) and stay within the same state with $P = 1/3$.

Algorithm 1: Model-driven bursty gap prediction

```
   Input: $\mathbf{x} = \{x^i\}_{i \in [1,m]}$: history bursty gap samples;
          $T_{last}$: Last bursty gap /*For prediction */
   Output: $T_{lb}$: $p-$lower confidence bound of next gap
1  /* Off-line M-GMM model training*/
2  while $k < k_{max}$ do
3      Train parameters $\boldsymbol{\phi^k}, \boldsymbol{\mu^k}, \boldsymbol{\sigma^k}$ with EM algorithm
4  $\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma}$ = $\underset{k}{\operatorname{argmin}}\ BIC(\boldsymbol{\phi^k}, \boldsymbol{\mu^k}, \boldsymbol{\sigma^k})$
5  foreach $i, j \in |S|$ do
6      $P_{ij} = \frac{s_{ij}}{\sum_{j \in [1,k]} s_{ij}}$
7  /* Online bursty gap prediction*/
8  $i = \underset{i}{\operatorname{argmax}}\ P(s_i = i | T_{last}, \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$
9  while $T_{lb} < \max\{\mathbf{x}\}$ do
10     if $\sum_{j=[1,k]} \phi_j * F_{\mathcal{N}(\mu_j, \sigma_j)}(T_{lb})$ > $p$ then
11         break
12     else
13         $T_{lb} = T_{lb} + \delta$
14 return $T_{lb}$
```

M-GMM Modeling and Training We now formally describe our proposed Markovian-Gaussian Mixture model. We denote the state space of bursty gap duration as $\mathcal{S} = \{s_1, s_2..s_k\}$, where k is the total number of states, each representing a Gaussian distribution. We use $\boldsymbol{\mu} = \{\mu_j\}_{j \in [1,k]}$ and $\boldsymbol{\sigma} = \{\sigma\}_{j \in [1,k]}$ to denote the mean and standard deviation vector of k distributions. Further, we use $\boldsymbol{\phi} = \{\phi_j\}_{j \in [1,k]}$ to denote the probability weight vector of k states, i.e, $\sum_{j \in [1,k]} \phi_j = 1$. This means that each bursty gap sample x_i is generated by first choosing a state s_i (according to $\boldsymbol{\phi}$), and then drawn from one of the k Gaussian models. The transition between two consecutive states is a Markov process and we use $\mathbb{P} = \{P_{ij}\}_{k*k}$ to denote the transition matrix, where $P_{ij} = P(s^j|s^i)$, i.e, the transition probability between states s^i and s^j.

The complete algorithm for training the parameters of M-GMM model from historical bursty gap samples $\mathbf{x} = \{x^i\}_{i \in [1,m]}$ is presented in Alg.1 (lines 1 to 6). We use the classical Expectation-Maximization (EM) algorithm to estimate the parameters $\boldsymbol{\phi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ of the Gaussian Mixture Model. A barrier in using EM is that we do not know the state number k, which is required before training the model. To solve this problem, we use the observation that there are only few (<10) clusters of bursty gaps. Thus, we set a maximum cluster number k_{max} and train k_{max} on different models. Then, we evaluate the goodness of fit for each model and pick the best one according to the Bayesian information criterion (line 4) . Finally, we obtain the Markov transition matrix by computing $P_{ij} = \frac{s_{ij}}{\sum_{j \in [1,k]} s_{ij}}$, where s_{ij} is the number of transitions from state s_i to s_j (line 6).

On-line Bursty Gap Prediction The M-GMM model is used to predict the $p-$lower confidence bound T_{lb} of the next bursty gap (Alg.1, lines 8-14), which is defined as $P(T_{lb} > t) \leq p$. We predict the lower confidence bound instead of absolute values because T_{lb} gives the probability that the actual bursty gap is longer than our prediction. It allows controlling the trade-off between the prediction accuracy and the bursty gap utilization using parameter p (see Sec.VI). First, we get the state i of the last bursty gap T_{last} through maximum likelihood estimation (line 8). Second, we gradually increase the values of T_{lb} in small steps (δ) until the average probability of $P(T_{lb} > t)$ exceeds p (lines 9-14).

B. Scheduling beam-searching and AP association

Both AP association and beam-searching operations must be completed within deadline T_{gap} to avoid disruptions to end-to-end connections, which can negatively impact the TCP performance [9]. We now formally define the deadline-aware scheduling problem as follows: Let $\mathcal{A}$ denote the set of all available 60GHz APs. N_{as} and N_{be} denote number of all associated APs, and the APs with fine-grained beam-forming, respectively. Thus, we have $N_{be} \leq N_{as} \leq |\mathcal{A}|$ where $|*|$ denotes the cardinality. Let T_{as} and T_{be} be the random time for associating with a new AP and/or beam-training latency. Thus, the scheduling problem can be formulated as follows:

$$\begin{aligned} \text{maximize} \quad & N_{be} \\ \text{subject to:} \quad & N_{be} \leq N_{as} \leq |\mathcal{A}| \\ & \sum_{i=1..N_{be}} T_{be}^{i} + \sum_{i=1..N_{as}} T_{as}^{i} < T_{gap} \end{aligned}$$

Since we only maximize N_{be}, i.e., APs with well-trained beams, the above problem is solved by greedily maximizing N_{be}. This simple logic is the core of the job scheduling process described as follows:

Step1: Evaluate the connection qualities (e.g., by checking RSSI) with each associated AP to create a candidate set.

Step2: Initiate beam-searching for associated APs with unsatisfactory beam-orientations (e.g., MCS rate < 385Mbps). Following this, attempt connections to additional APs.

Step3: Update the empirical average estimation of T_{be}, T_{as}. Return to step 2 before the end of T_{gap}.

Through the above steps, MAP establishes connections to surrounding APs with freshly trained beams. However, we note that end-to-end performance is a function of other factors such as AP load, which cannot be captured at the physical/link layer alone. MAPs solve this problem by coupling exploration-exploitation based path selection and flexible flow scheduling, as described in the next section.

IV. Multi-Path Flow Controller

We first provide a high-level operational overview of 60GHz multi-path scheduling as follows. An initial *control sub-flow* is initiated between the client and server on the sub-6GHz band. While this control connection can be used for data transmission, its main task is to establish and maintain the overall TCP connection. This leverages the advantage of wide-coverage of lower frequency band, and will guarantee that a *network change error* is not triggered even if all 60GHz links are disrupted. Then, the path manager opens a new *data transfer sub-flow* on *one link* associated with the selected 60 GHz AP that can potentially provide the best performance, while other paths remain in back-up mode. We will describe how to switch data transmission sub-flows to the best AP in the following Sec.IV-A.

If the current 60GHz link is blocked, a new sub-flow on an alternative 60GHz path is immediately initiated, and the previous sub-flow on the impacted link is stopped. This process is near-instant since the client is already associated with several back-up 60GHz APs through periodic beam-searching process (Sec.III-B). After transmitting one chunk of data, the path manager evaluates the performance of the selected path for the active sub-flow. It may decide to stay with the current path or switch to a different one (i.e., activate the flow to a different AP). This continuous decision prevents a client from being locked to a single 60GHz link with degraded performance due to blockage or antenna misalignment.

Algorithm 2: UCB-M based AP selection

```
Input: $\mathcal{A}$: Available AP candidates set; $v_i$: reward of last
       selected AP $i$ ;
Output: $j$: Next selected AP
$v_i = v_i / v_i^{max}$ /*Normalize reward*/
Update $\widehat{v}_i$ with new $v_i$ according to (2)
foreach $j \in \mathcal{A}$ do
    if $T_j(n) = 0$ then
        return j
$j = \arg\max_{j \in \mathcal{A}} \widehat{v}_j + \sqrt{\frac{2 \ln n}{T_j(n)}}$
return $j$
```

A. AP Selection Algorithm

In this section, we describe the decision process for identifying the best path, which is the key problem the flow controller has to solve. This is an AP assignment problem that finds AP j^* such that $j^* = \arg\max_{j \in \mathcal{A}} Q_j$, where $\mathcal{A}$ is the available AP candidate set and $Q(*)$ is the quality evaluation function of a specific AP, which we explain below.

Making this selection is a difficult problem for following two reasons:

- The suitability of the 60GHz AP is decided by several factors including channel conditions, random blockage, and load on AP, which are unknown at the client side and hard to predict before data transmission.
- The above factors can be time-varying (e.g, load) and location dependent (e.g blockage probability), which lead to variability in the 60GHz AP performance. Naively selecting an AP according to one-time performance sampling does not work well since the link quality can quickly change in a 60GHz network.

Given these challenges, we cast the AP selection problem into multi-armed bandit framework, formulated as an exploration-exploitation process. Specifically, consider each AP is associated with an unknown *reward* distribution, which reflects the random downlink performance it can provide for certain client. We base our approach on the upper confidence bound (UCB) policy. The key idea of UCB is to select an AP that has the highest empirical upper confidence bound on its performance. Compared with other policies, UCB does not explicitly need the weight of exploration vs. exploitation (e.g., ϵ - Greedy, SoftMax), nor does it need a priori assumption of the reward distribution (e.g., Gittins index, Thompson Sampling).

1) Defining the reward for a chosen path: Consider the following two performance metrics: average downlink data rate r and link stability s. The average rate r is computed by dividing the bytes of each bursty chunk by the transmission time of that chunk. Link stability s characterizes how often the connection is interrupted (due to blockage or beam misalignment).The more frequently MAP switches the path when choosing certain AP, the less stable that AP is, and as a result, the AP earns lower rewards.

The reward for AP j can now be defined as $v_j = r_j * s_j$. If AP j is selected for $n'th$ bursty chunk, a new reward sample of a can be written as:

$$v_j(n) = r_j(n)(1 - \frac{T_j^s(n)}{T_j(n)}), \tag{1}$$

where $r_j(n)$ is average download speed of $n'th$ chunk. $T_j(n)$ is the total number of times that AP j was selected and $T_j^s(n)$ is the total number of path switching times during the data transmission.

2) Modified UCB algorithm with moving average : Classical UCB simply picks the AP that maximizes $Q_j(n)$ as $Q_j(n) = \overline{v}_j(n) + \sqrt{\frac{2 \ln n}{T_j(n)}}$. Here, $\overline{v}_j(n)$ is the empirical mean reward for AP j until the $n'th$ selection, which is computed by $\sum_{i=1}^{T_j(n)} v_j(i)/T_j(n)$. The extra term $f_j(n) = \sqrt{\frac{2 \ln n}{T_j(n)}}$ can be regarded as the uncertainty level of the reward. Note $f_j(n)$ is inversely proportional to $T_j(n)$, i.e., the less AP j is selected, the more uncertainty persists about its actual performance. The combined metrics $\overline{v}_j(n)$ and $f_j(n)$ define the upper confidence bound of the reward given to AP j.

UCB's responsiveness to the varying AP performance can be slow as it takes cumulative average reward from all historical samples when computing $Q_j(n)$. To solve this problem, we use Exponential Weighted Moving Average (EWMA) $\widehat{v}_j(n)$ instead of cumulative average for $Q_j(n)$:

$$\widehat{v}_j(n) = \begin{cases} v_j(n) & \text{if } n = 1 \\ \alpha * v_j(n) + (1 - \alpha) * \widehat{v}_j(n-1) & \text{otherwise.} \end{cases} \tag{2}$$

We use UCB-M to denote the UCB algorithm combined with the EWMA reward. EWMA discounts the reward samples taken in the remote past, thus allows the algorithm to follow the varying trend of reward distribution more closely. Here α is the average factor within range (0, 1]. As $\alpha \to 0$ the EWMA approximates to the cumulative average while $\alpha \to 1$ means simply taking the instantaneous reward sample without averaging.

V. Convergence Analysis of UCB-M

In this section, we provide theoretical analysis that the UCB-M algorithm achieves nearly optimal performance by limiting the selection of sub-optimal APs only logarithmic number of times with an appropriate moving average factor α.

We first define the following notations. Assume there are K available APs in total. $\mu_j = \mathbb{E}[v_j(n)]$ mean reward of AP j. We use μ^* to represent the highest mean reward of all APs, such that $\mu^* = \max_j \mu_j$. We define the *regret* as the $\Delta_j = \mu^* - \mu_j$, which can be interpreted as the difference in the rewards obtained as a result of the suboptimal AP selection vs. picking the optimal one. Thus the expected regret after n iterations is:

$$R_n = \mathbb{E}[n\mu^* - \sum_j T_j(n)\mu_j] = \sum_{j=1}^{K} \mathbb{E}[T_j(n)]\Delta_j \tag{3}$$

The optimal algorithm for exploration-exploitation can only achieve logarithmic regret over increasing iterations [10]. The following theorem establishes the conditions under which the UCB-M algorithm returns such a logarithmic regret.

Theorem1: *For an arbitrary reward distribution, the expected regret after any number n of iterations is $O(\frac{K \ln n}{\Delta})$ as long as $T_j(n) \leq \frac{4}{\alpha} - 2$. Δ is the minimal regret of suboptimal APs.*

Proof: Our proof was based on the framework of [11]. The key idea is to show that the probability of selecting the non-optimal AP decays quickly with increasing n. If we select the non-optimal AP j at time t, then,

$$\widehat{v}_j(t) + f_j(t) > \widehat{v}^*(t) + f^*(t) \tag{4}$$

where $\widehat{v}^*(t)$ denotes the reward from the selection of the optimal AP at time t. If (4) holds, then *at least* one of the following three conditions must be true:

$$\widehat{v}_j(t) \geq \mu_j + f_j(t) \tag{5}$$

$$\widehat{v}_j^*(t) \leq \mu^* - f^*(t) \tag{6}$$

$$\mu^* \geq \mu_j + 2 * f_j(t) \tag{7}$$

Thus, to bound the probability of (4), we compute the probability of (5), (6) and (7). We first bound (5) and (6) by introducing the following lemma:

Lemma1: *Let $X_i, i \in [1, n]$ be random variables with range $[0, 1]$ and mean $\mathbb{E}(X_i) = \mu$. Let S_n be the EWMA of $X_1, \ldots, X_n$, $n > 1$ with parameter α. Then, for $t \geq 0$, the following inequality holds:*

$$P\{S_n - \mu \ \geq t\} \leq e^{-2t^2\left(\frac{2}{\alpha}-1\right)} \tag{8}$$

Proof: It can be trivially proven that the mean of the moving average is equal to the mean of the original random variable, i.e., $\mathbb{E}[S_n] = \mu$. To computer the probability $P\{S_n - \mathbb{E}[S_n] \geq t\}$, we write the moving average S_n as $S_n = \alpha \sum_{i=0}^{n-1}(1-\alpha)^i X_{n-i}$. While S_1 here is expressed in a slightly different format from our definition in (2), this notation simplifies the discussion without any influence on the conclusion. Then, we have:

$$\begin{aligned} P\{S_n - \mathbb{E}[S_n] \ \geq t\} &= P\{e^{s(S_n - \mathbb{E}[S_n])} \geq e^{st}\} \\ &\leq e^{-st}\mathbb{E}[e^{s(S_n - \mathbb{E}[S_n])}] \\ &= e^{-st}\prod_{i=0}^{n-1}\mathbb{E}[e^{sX(1-\alpha)^i(X_{n-i} - \mathbb{E}[X_{n-i}])}] \\ &\leq e^{-st}\prod_{i=0}^{n-1} e^{\frac{s^2\alpha^2(1-\alpha)^2}{8}} \\ &\leq e^{-2t^2(\frac{2}{\alpha}-1)} \end{aligned} \tag{9}$$

Here, the first inequality comes from the Markov inequality and the second inequality comes from the Hoeffding Lemma.

Lemma 1 bounds the probability that the EWMA deviates from the mean value of the random variable. With Lemma 1, we can bound the probability of (5) as follows:

$$P\{\widehat{v}_j(t) \geq \mu_j + f_j(t)\} \leq e^{-2f_j^2(t)(\frac{2}{\alpha}-1)} \leq n^{-\frac{2}{T_j}\left(\frac{2}{\alpha}-1\right)} \tag{10}$$

Defining $\beta = \frac{4 \log n}{\Delta_j^2}$ and if $T_j \geq \beta$, (7) must be false since $\mu_j + 2 * f_j(t) \leq \mu_j + 2 * \sqrt{4 \ln n / \Delta_j^2} \leq \mu_j + \Delta_j = \mu^*$

We bound T_j through its conditional expectation :

$$\mathbb{E}[T_j] = P(T_j \leq \beta)\mathbb{E}[T_j | T_j \leq \beta] + P(T_j \geq \beta)\mathbb{E}[T_j | T_j \geq \beta] \tag{11}$$

We need to bound different items of (V) separately. First it is trivial to see $P(T_j \leq \beta) \leq 1$, $\mathbb{E}[T_j | T_j \leq \beta] \leq \beta$ and $\mathbb{E}[T_j | T_j \geq \beta] \leq n$. Besides, we know $T_j \geq \beta$ happens when both (5) and (6) are violated, thus $P(T_j \geq \beta) \leq n^{-\frac{2}{T_j}(\frac{2}{\alpha}-1)}$ (from (10)).

$$\mathbb{E}[T_j] \leq \frac{4 \log n}{\Delta_j^2} + 2n^{-\frac{2}{T_j}\left(\frac{2}{\alpha}-1\right)+1} \tag{12}$$

We now write expected regret as:

$$\begin{aligned} R_n &= \sum_{j=1}^{K} \mathbb{E}[T_j(n)]\Delta_j = \sum_{j=1}^{K} \frac{4 \log n}{\Delta_j} + \sum_{j=1}^{K} \Delta_j * 2n^{-\frac{2}{T_j}\left(\frac{2}{\alpha}-1\right)-1} \\ &\leq \frac{4K \log n}{\Delta} + 2K\Delta n^{-\frac{2}{T_j}\left(\frac{2}{\alpha}-1\right)+1} \end{aligned} \tag{13}$$

Note that since $T_j(n) \leq \frac{4}{\alpha} - 2$, $-\frac{2}{T_j}\left(\frac{2}{\alpha} - 1\right) + 1 \leq 0$. So $R_n \sim O(\frac{4K \log n}{\Delta})$. Thus we complete the proof.

An implication of Theorem 1 is that to achieve logarithmic regret bound, the α should be as small as possible to ensure the $T_j(n) \leq \frac{4}{\alpha} - 2$ holds. However, setting α too small also makes UCB-M react slower to the dynamic 60GHz links. To quantify this tradeoff, we present following theorem:

Theorem 2: *Let $X_1, \ldots, X_n$ be a sequence of random variables such that $E(X_1) = E(X_2) = \ldots = E(X_k) = \mu_1$*

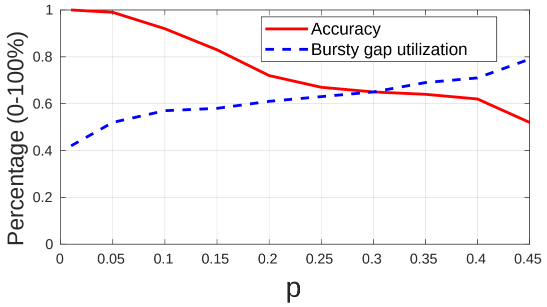


Fig. 7: Trade-off between accuracy and bursty gap utilization.

and $E(X_{k+1}) = E(X_{k+2}) = \ldots = E(X_n) = \mu_2$. S_n *is the EWMA of* $X_1, \ldots, X_n$. *Then* S_n *converges to* μ_2 *in mean for sufficiently large* n.

Proof: This is proved by expressing S_n as follows:

$$S_n = \alpha * \sum_{0}^{n} (1-\alpha)^{n-i} X_i \tag{14}$$

Thus we have:

$$\mathbb{E}(S_n) = \alpha * (\sum_{i=0}^{k} (1-\alpha)^{n-i} \mu_1 + \sum_{k+1}^{n} (1-\alpha)^{n-i} \mu_2) \tag{15}$$

Computing the sums of the geometric series and substituting into $|\mathbb{E}(S_n) - \mu_2|$, we get:

$$\begin{aligned} |\mathbb{E}(S_n) - \mu_2| &= |(1-\alpha)^{n-k}(\mu_1 - \mu_2) - (1-\alpha)^n * \mu_1| \\ &\leq (1-\alpha)^{n-k}|\mu_1 - \mu_2| + (1-\alpha)^n * |\mu_1| \end{aligned} \tag{16}$$

When $n \to \infty$, both $(1-\alpha)^{n-k}|\mu_1 - \mu_2|$ and $(1-\alpha)^n * |\mu_1|$ goes to zero. So $|\mathbb{E}(S_n) - \mu_2| \to 0$.

From the proof of Theorem 2 we could see that the smaller α is, the slower S_n will converge to the true mean value of the changed reward distribution μ_2 and the longer it will retain the bias estimation of X_2. On the other extreme, $\alpha = 1$ yields $|S_n - \mu_2| = 0$, which means the moving average will instantaneously keep up with the varying reward distribution.

VI. Performance Evaluation

We validate MAP with a (i) real-world video streaming application to measure the effectiveness of the bursty gap prediction model, (ii) trace-driven Linux emulation testbed with commercially available 60GHz APs and laptops serving as mobile clients, and (iii) macro-scale simulation for dense urban scenarios.

A. Accuracy of model prediction

We first measure the accuracy of our model by studying the instances when the actual bursty gap T_{gap} is observed to be unexpectedly *shorter* than the lower confidence bound T_{lb} that we predicted. We also study the bursty gap utilization, which is defined as the fraction of the link idle time that can be actually utilized for AP association and beam-searching:

$$U = \frac{\sum_{i=1}^{m} [T_i^{lb} * \mathbb{1}\{T_{lb}^i < T_{gap}^i\} + T_{gap}^i * \mathbb{1}\{T_{lb}^i < T_{gap}^i\}]}{\sum_{i=1}^{m} T_{gap}^i}$$

. where m is the size of dataset used for the study and $\mathbb{1}$ denotes the indicator function.

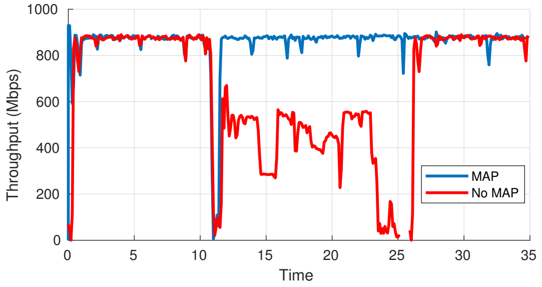


Fig. 8: Fast link performance recovery with MAP.

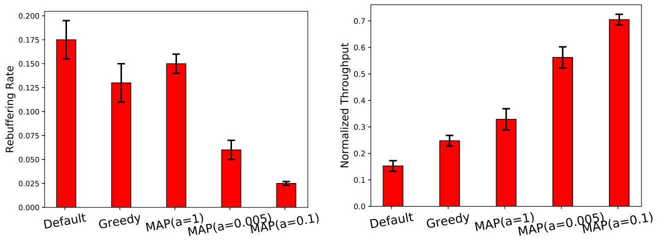
From Fig. 7, we see that the parameter p controls the trade-off between accuracy and the bursty gap utilization. Larger p encourages aggressive use of the channel by increasing T_{lb}. While this leads to better bursty gap utilization (up to 80%), the tradeoff involves sacrificing the prediction accuracy. On the other hand, lower p returns a more conservative prediction of the link idle time, yielding higher accuracy yet worse bursty gap utilization. As a baseline, the arithmetic mean predictor is at one extreme end on the scale, with about 50.2% accuracy with 74% bursty gap prediction. In comparison, there are two advantages of our predictor: First, the M-GMM model captures the distribution and transition probability of the bursty traffic pattern better than a naive model, thus allowing it to adjust the estimation appropriately according to the varying states of bursty gap. Second, the tunable value p allows flexible control over the trade-off between the prediction accuracy and the time budget for lower layer operations (AP association and beam-searching). For example, users in a high density 60GHz network can choose larger p to better utilize the bursty gap to associate with more APs.

B. Trace-driven testbed evaluation

We collect fine-grained bandwidth traces (at the millisecond level) by saturating the 60GHz link with `Iperf` traffic between the Talon AD7200 router and Acer laptops placed at various locations in the lab. To further emulate practical situations, we introduce random blockage with people walking in the vicinity and limited client motion.We implement a network emulator based on `tc` tool-suite in Linux for on-demand replaying of the channel characteristics.

1) Proactive and seamless link recovery: In the classical approach, a client remains associated with its current AP, even after the throughput drops significantly (see Fig. 8). We show the performance gain when we manually force it to switch to the alternative AP at the 25s mark. However, the original connection is lost during the AP switching due to the network change error, which further prolongs the link recovery time and degrades the client's QoE. As opposed to this, MAP fully recovers its throughput in about 600ms. The speed-up in the link recovery is because MAP is already associated with backup APs with better LoS connections.

2) Impact on user QoE for uncompressed video streaming: We next select a real-world bandwidth intensive application: a video server that transmits uncompressed video frames (1920P*1080P, 24fps) to the client. Theoretically, it takes 760 Mbps to transmit the video without buffering, which is easily



(a) (b)

Fig. 9: Comparison of (a) Rebuffering rate and (b) average bitrate comparison. The bitrate is normalized between the theoretical lower and upper bound ([550Mbps, 850Mbps]) computed from traces.

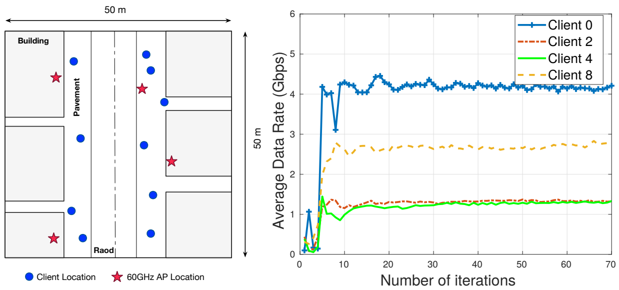
supported by a 60GHz link with LoS connection. The video stream is split into multiple chunks (of 2s each) and the video buffer can save up to 5 video chunks.

We focus on evaluating two performance metrics: (i) *Rebuffering rate:* defined as fraction of total buffering latency to video length, and (ii) *Average bitrate:* calculated as average transmission throughput across all chunks. Besides, we compare MAP with following two baseline strategies: (a) *Default:* in which the client connects to AP with highest RSSI and (b) *Greedy selection:* where the client greedily chooses an AP that historically has the best throughput performance.

As shown in Fig. 9(a), MAP ($\alpha = 0.1$) achieves up to 7X and 5X lower rebuffering rate than both the default and the greedy schemes, respectively. Also, from Fig. 9(b), we see that MAP ($\alpha = 0.1$) also attains much higher (5X and 2X) normalized average bit-rate than these two schemes. This is because MAP quickly converges to the AP with optimal performance, which greatly improves its resilience to highly fluctuating 60GHz link conditions. As comparison, the QoE of video streaming application suffers in classical architectures as the client is unable to seamlessly switch to a different 60GHz AP even when the connection quality drops significantly.

Interestingly, greedy selection of the AP with the highest historical performance results in worse QoE than the default baseline. The main reason is that the greedy scheme suffers from a *prediction bias*, as the client only has fresh observations of its currently connected AP, while historical measurements of other APs are likely to become outdated due to the inherent variability of the 60GHz links. For example, the sudden performance drop of the optimal AP may force client to revert to a sub-optimal AP with much lower throughput, even though the the performance of original AP recovers quickly after the next beam-searching phase. In comparison, MAP overcomes this issue by constantly exploring opportunities connecting with better APs over time.

Finally, appropriate setting of the averaging parameter α has an impact on the performance of MAP. At one extreme, choosing $\alpha = 1$ makes the performance of MAP approach that of the greedy selection. On the other extreme, choosing a very low α (e.g, 0.005) makes MAP insensitive to the changing networking conditions. This matches our previous observations drawn from our theoretical analysis (see Sec. V). In general, $\alpha = 0.1$ represents an ideal trade-off between the ability to adapt to varying network conditions and also being resilient to outages in the 60GHz link.



(a) (b)

Fig. 10: (a) Urban area simulation scenario; (b) UCB-M achieves fast convergence for each client (shown for 4 clients).

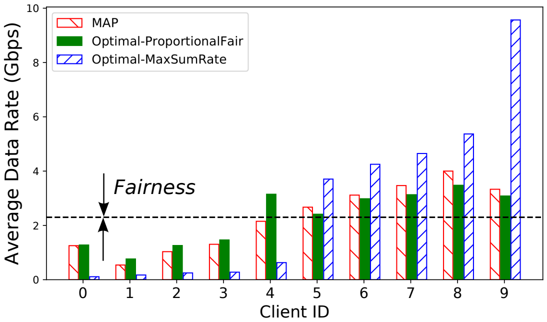


Fig. 11: MAP achieves better proportional fairness through UCB-M

C. System level urban area simulation

We use a trace-driven simulation of an urban street with multiple surrounding APs to study MAP's performance in a large-scale network. An important issue that we address is: whether MAP converges and achieves fairness in resource allocation, as the underlying UCB-M algorithm is a selfish mechanism that optimizes the client's own performance without considering the other clients sharing the same AP.

1) Scenario description and metrics: Our typical urban environment is composed of 4 APs deployed on each side of the street for serving 10 randomly distributed users (Fig. 10(a)). The maximum transmission power of each AP is 36 dBm, carrier bandwidth is 2.16 GHz and the thermal noise level is -97dBm/Hz. We use the channel model in [7] that specifies three states of the 60GHz channel availability (LoS, NLoS and Outage) with respective probabilities . We assume each AP uses round robin scheduling. Thus, each time-frequency resource block is equally allocated to each connected client served by an AP [7]. We focus on evaluating the per-user bitrate (in bps) computed via the Shannon limit $R = \frac{BW}{N}\log(1+SINR)$, where BW is the AP bandwidth and N is the total number of clients connected with that AP.

2) Convergence speed: Fig. 10(b) shows the convergence behavior of MAP for 4 selected clients. In general, MAP converges rapidly despite the performance fluctuation in the first 10 iterations. This because UCB-M attempts every unexplored path, which results in fluctuations in the client performance when it selects a sub-optimal path. However, the client reaches stable data rate in about 15 iterations.

3) Near-optimal bandwidth allocation: While above experiments show the fast convergence behavior of MAP, whether the final AP selection is optimal or not remains unknown. For this, we compute the global optimal AP assignment under perfect knowledge of the network (AP/client number and locations, path loss, blockage probability) by modeling it as a Mixed-Integer Non-Linear (MINLP) Problem . Since MINLP is NP-hard, we solve it with the Bonmin toolbox, which produces a solution based on branch-and-bound method [12].

We compare MAP to the globally optimal solutions computed with two objectives: (i) *Max sum rate:* that aims at maximizing the total user data rate, and (ii) *Proportional Fairness:* obtained by maximizing the sum of the logarithms of per-user data rates.

Fig.11 presents the average bit-rates of clients 1-10. While sum rate maximization does achieve higher total user bit-rate, it leads to extremely unfair bandwidth allocation, i.e., 50% of the clients receive very high throughput while others are near-starved. In comparison, the proposed UCB-M algorithm achieves very close performance with the global optimal proportional fairness, despite UCB-M being a distributed algorithm that works only on client side with no global information about network conditions. The main reason for this performance is that UCB-M schedules the flows away from congested AP for better performance, thus enabling clients to balance the load on APs.

VII. Related Work

We survey the most relevant protocol research in this section.

60GHz link profiling: [3] studies indoor 60GHz link layer performance with a specific focus on the impact of blockage, the overhead of beam-searching and AP discovery latency. [2] studies the upper layer performance of the 60GHz network, with extensive evaluation in outdoor environment and concludes that today's commercial NICs are able to adjust the beam-directions to overcome intermediate blockages. Our paper extends beyond measurements, and also describes a solution for better utilizing multiple 60GHz AP deployments.

PHY/MAC innovations for resilient links: Many works increase user-resiliency to blockages in the 60GHz links via efficient beam-searching through programmable reflector, ray tracing, and exploiting the spatial correlations of 60GHz channels [4]–[6]. [13] proposes using 2.4/5GHz band to assist the millimeter-wave beam-searching by inferring the LoS direction between the communicating devices from omni-directional 2.4/5GHz signals. This out-of-band information greatly reduces the need for extensive beam-searching, and thus enables an efficient antenna sector selection. We note that any of the above may be used with MAP (if integrated in the wireless standard) as the latter only defines the best durations to execute these functions.

Multi-AP association and selection: Multi-AP association can be effectively managed with a global view of the network scenario (such as the number and distribution of AP/clients, AP load/capacity, channel conditions etc.), using optimization problems [14], [15]. In comparison, MAP operates locally in a client without cooperation of the AP and yet exhibits near-optimal proportional fairness in resource allocation.

Multi-path TCP: MPTCP is an active research area. MAP is partly inspired by [16], which aims at seamless handover in legacy IEEE a/b/g/n networks. However, the focus in MAP is completely different: it optimizes the use of directional links at 60GHz, instead of addressing the low CSMA/CA efficiency caused by the hidden terminal problem in legacy bands.

VIII. Conclusion

MAP achieves multi-AP association by leveraging information across the protocol stack and triggering functions mainly at transport and phy/link layers, without modifying the existing standards. Through implementation in commercially available 60GHz devices and trace-based emulations, we have demonstrated the benefits of MAP in achieving 5-7x reduction in re-buffering rate in HD video streaming, rapid convergence to the best possible AP given the frequent fluctuations in the quality of the 60GHz links, and fair allocation of resources among a number of clients. Future research will focus on multi-NIC scenario with concurrent active transmissions to multiple APs.

Acknowledgment

This work was supported by the U.S. Office of Naval Research under grant number N00014-16-1-2651.

References

[1] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "Ieee 802.11 ad: directional 60 ghz communication for multi-gigabit-per-second wi-fi," *IEEE Communications Magazine*, 2014.

[2] Y. Zhu, Z. Zhang, C. Marzi, U. Madhow, B. Y. Zhao, and H. Zheng, "Demystifying 60ghz outdoor picocells," in *ACM MobiCom*, 2014.

[3] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 ghz indoor networking through flexible beams: A link-level profiling," in *ACM SIGMETRICS*, 2015.

[4] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality." in *NSDI*, 2017.

[5] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "Beamspy: Enabling robust 60 ghz links under blockage." in *NSDI*, 2016.

[6] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 ghz networks via model-driven beam steering," in *INFOCOM 2017*.

[7] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, and R. W. Heath, "Modeling and analyzing millimeter wave cellular systems," *IEEE Transactions on Communications*, 2017.

[8] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," in *ACM IMC*, 2012.

[9] A. K. Al-Ali and K. Chowdhury, "Tfrc-cr: An equation-based transport protocol for cognitive radio networks," *Ad Hoc Networks*, 2013.

[10] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, 1985.

[11] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, 2002.

[12] "Bonmin," https://projects.coin-or.org/Bonmin.

[13] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: mm-wave beam steering without in-band measurement," in *IEEE INFOCOM*, 2015.

[14] S.-C. Lin and I. F. Akyildiz, "Dynamic base station formation for solving nlos problem in 5g millimeter-wave communication," in *IEEE INFOCOM*, 2017.

[15] O. B. Karimi, J. Liu, and J. Rexford, "Optimal collaborative access point association in wireless networks," in *IEEE INFOCOM*, 2014.

[16] A. Croitoru, D. Niculescu, and C. Raiciu, "Towards wifi mobility without fast handover." in *NSDI*, 2015.