Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 175 (2020) 365–372

Procedia Computer Science

www.elsevier.com/locate/procedia

The 15th International Conference on Future Networks and Communications (FNC)
August 9-12, 2020, Leuven, Belgium

MinDFul: Using double links for stabilizing mmWave wireless channels for application to autonomous vehicles and augmented reality

Hossein Ajorloo *[a], Cormac J. Sreenan[a], Roberto Bomfin[b], Martin Danneberg[b], Gerhard Fettweis[b]

[a] *School of Computer Science and IT, University College Cork, Cork, T12 XF62, Ireland*
[b] *Vodafone Chair Mobile Communication Systems, Technische Universität Dresden, 01062 Dresden, Germany*

Abstract

Applications that require short-range ultra-high bitrate communication, such as cable removal in virtual reality games and communication between autonomous vehicles, are examining solutions such as millimetre wave wireless (mmWave). When using mmWave, steerable directional antennas are used to mitigate the severe signal power attenuation common with high frequencies. Nonetheless, even small movements in the user device can cause a sudden drop in data-rate down (even to 0 bits/s) making mmWave channels unstable and unusable. To make the channel more stable for the aforementioned applications, which are vulnerable due to frequent blockages and fast movement, we designed and developed a robust solution based on a double link mmWave system. We duplicate the radio transceivers (RT) of a user device (UD) to increase the probability of finding line of sight to an access point (AP) representing the other side of the communication channel. The AP selects one RT of the UD for communication, based on continuous measurement of quality compared to the channel of the other RT. This concept was implemented in a laboratory environment and evaluated using a series of controlled experiments. The experiments serve to validate that using double links is feasible, and is considerably more robust and it can double the link utilization, compared to only using one mmWave link. These results show great promise for the concept, by demonstrating that using multiple mmWave links yields ultra-high bit-rate wireless communication with no disruption, even in the presence of blockages and mobility.

Keywords: Millimetre wave (mmWave); Scheduling algorithm; Virtual reality (VR) games; Autonomous vehicles; Wireless communication; Beamsteering protocol; Steerable antenna array; Real-time system; Experimental testbed; 5G;

* Corresponding author. Tel.: +353 (21) 420-5912.
E-mail address: hossein.ajorloo@ucc.ie (Hossein Ajorloo *).

10.1016/j.procs.2020.07.052

1. Introduction

With the advance of technology, the number of applications demanding ultra-high bitrate communication increases rapidly. Among all such applications, cable removal in augmented/virtual reality (AR/VR) games and inter-vehicle and vehicle-to-infrastructure communication in autonomous (driverless) vehicles are two very demanding applications. In AR/VR games, the communication between the game console and the headset requires +6 Gbps data-rate which is commonly provided with cables [1]. Tethering using a cable is a barrier for players, preventing them from freely moving around and can lead to falls and injuries. On the other hand, communication between autonomous vehicles requires 3 - 40 Gbps data-rate to be able to share the vehicles sensor data [3].

mmWave is the range of frequencies between 30-300 GHz which can provide +10 Gbps data-rate thanks to the available wide bandwidth [2]. However, unlike sub-6GHz wireless channels where a constant throughput with the maximum data-rate can be provided for a long-time even when the devices are moving or there are no line-of-sight (LoS) path available between them, the mmWave channel is not stable. To alleviate this problem, in this paper we propose to use two mmWave links, one as the auxiliary for the other. We call this method "MinDFul": **M**illimetre-wave **Lin**k **D**oubling **Ful**l-stack experiments. This paper has three contributions:

1. Design and development of a scheduling algorithm to control throughput of a User Device (UD) with two Radio Transceivers (RTs) by the Access Point (AP).
2. Implementing double-link communication on a laboratory testbed.
3. Designing and conducting two sets of experiments to validate the approach and evaluate performance for blockage and movement scenarios.

The rest of this paper is organized as follows: in section 2 we state the problem that is addressed. Section 3 presents the design and implementation of our solution. Section 4 explains our experimental methodology, followed in section 5 with the experimental results. The related work is summarized in section 6. Finally, section 7 concludes the paper.

2. Problem Statement and Application

To provide ultra-high bit-rate wireless communication for applications such as AR/VR hardware and communication between autonomous vehicles, emerging mmWave wireless technology is an attractive choice. A key problem with this technology is the instability of the channels caused by blockage or movement of the communicating devices. One solution is to use steerable directional antennas. The antenna beam steering protocols are designed in such a way that they can steer antenna beams in less than a millisecond. So, if the devices move and they lose their wireless channel, they find another pair of antenna beams on which they can communicate (if such pair of beams exist). If LoS between the antennas are blocked (which can happen even with small objects such as a human hand), and if a Non-LoS path exists between another pair of beams, the system can switch to those beams. But, mmWave electromagnetic waves do not show good reflection properties. So, it is not reasonable to assume that a non-LoS path is available.

Another problem is the fact that mmWave antennas are unable to provide enough signal strength from their rear side, which means that the devices are unable to communicate when their antennas do not face toward each other, even with their strongest modulation and coding scheme (MCS). In fact, it has been observed that when the rear side of a mmWave antenna is forward facing, the throughput drops to almost zero [5, 8].

These problems prevent the practical use of mmWave technology for AR/VR and autonomous vehicles applications. In this paper, we propose to use two sets of RTs for a device and to switch from one RT to the other one when the first one is blocked or its LoS path is lost. We implemented this idea on the online wireless testbed (OWL) and conducted complete sets of experiments to evaluate its performance. We upgrade the laboratory testbed used in [5, 8] to have double-link communication functionality and conducted two sets of experiments for blockage of a UD antenna and its movement causing it lose its LoS path.

To use this solution in the AR/VR games hardware, the manufacturers can install two mmWave RTs on the AR/VR headset, one in the left and one in the right side. Our proposed method in the next section will switch between RTs whenever LoS between one RT and the game console is blocked, or when the users moves their head or body making one of the RTs lose their LoS. It is less likely that a user blocks both RTs LoS paths simultaneously and also, it is less likely that a user's head is put in a position where both RTs which are mounted back-to-back with 180° angle to each other los their LoS with game console's antenna.

Similarly, we can mount multiple mmWave RTs all around an autonomous vehicle so that when one pair of RTs between two vehicle loss their LoS path, both or one of them can switch to another RT to keep communication going, this providing a highly robust solution.

In this paper, our aim is to demonstrate that using double links is feasible, and is considerably more robust and it can double the link utilization, compared to only using one mmWave link. Naturally, the use of an additional transceiver incurs some additional cost, which for the target application scenarios is negligible.

3. Double Link Solution: Design and Implementation

In this section, we present the design and implemention of MinDFul, the double link solution to make the mmWave channel more stable.

3.1. Solution Design

In almost all today's wireless networking protocols a form of time-division duplex is used, where the AP divides ongoing time into small fixed-size frames, each comprising N_s timeslots. Typically, two slot types are used: (1) probing slot for measuring the channel, transmitting control and feedback configuration, (2) data slot for actual data transmission. The AP decides on how many slots are assigned for probing and data transmission for each UD in the up-link (UL) and down-link (DL) directions, N_d slots of each frame are used for DL direction and $N_u = N_s - N_d$ slots are used for UL direction. The first slots in each direction are used for probing (mainly for antenna tracking) and the remaining slots are used for data transmission. To make the explanation of our method easier, here we continue with an example: Communication takes place between one AP and one UD where the UD has two RTs, designated RT0 and RT1. It is noteworthy to emphasise that we can use more than two RTs and make a more stable channel. Obviously it would have cost implications. Here we choose two RTs for validation but it can easily be generalised to more RTs.

We designed a scheduling algorithm to assign all data slots to the RT of the UD with a better Signal-to-Noise Ratio (SNR). The idea is straightforward: start by assigning all timeslots to RT0. When the average SNR of RT1 in both UL and DL directions goes above the average SNR of RT0 plus T_1 dB, then assign all timeslots to RT1, where T_1 is a threshold value. We continue communicating through RT1 until the average SNR of RT0 goes above the average SNR of RT1 plus T_0 dB, at which point all timeslots are reassigned back to RT0. T_0 is another threshold value. We first define the following variables:

$$\begin{aligned}\Delta_{AP} &= \overline{\mathrm{SNR}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}} - \overline{\mathrm{SNR}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT1}},\\ \Delta_{UD} &= \overline{\mathrm{SNR}}_{\mathrm{RT0}\leftrightsquigarrow\mathrm{AP}} - \overline{\mathrm{SNR}}_{\mathrm{RT1}\leftrightsquigarrow\mathrm{AP}},\end{aligned} \tag{1}$$

where $\overline{\mathrm{SNR}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$ and $\overline{\mathrm{SNR}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT1}}$ are the average SNR values measured by the AP over the signals received from RT0 and RT1 respectively, which are obtained by weighted averaging the measured SNR values on the last M frames. Similarly, $\overline{\mathrm{SNR}}_{\mathrm{RT0}\leftrightsquigarrow\mathrm{AP}}$ and $\overline{\mathrm{SNR}}_{\mathrm{RT1}\leftrightsquigarrow\mathrm{AP}}$ are the average SNR values measured by RT0 and RT1 respectively, over signals received from the AP. More precisely,

$$\overline{\overline{\mathrm{SNR}}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}} = \frac{\sum_{i=0}^{N-1} \alpha^i . \overline{\mathrm{SNR}}_{i,\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}}{\sum_{i=0}^{N-1} \alpha^i}, \tag{2}$$

where $0 < \alpha \leqslant 1$ is a number used for weighted averaging. $\overline{\mathrm{SNR}}_{i,\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$ denotes the equal-weight average SNR value of the ith previous frames. By setting $\alpha = 1$, we do equal-weigh or simple averaging. By setting a small value for α, e.g. $\alpha = 0.001$, we only consider the SNR value of the last frame. Any number in between results in weighted averaging. In Eq. (2),

$$\overline{\mathrm{SNR}}_{i,\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}} = \frac{1}{S_{i,\mathrm{RT0}}} \sum_{j=1}^{S_{i,\mathrm{RT0}}} \mathrm{SNR}_{i,j,\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}, \tag{3}$$

where $\mathrm{SNR}_{i,j,\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$ denotes the measured SNR at RT0 on data slot j of the previous ith frame. $S_{i,\mathrm{RT0}}$ denotes the number of data slots assigned to RT0 in the previous ith frame. The other average values are computed similarly.

Algorithm 1: Selection of RT0/RT1 for data-slot assignment in double-link communication.

---

input : Matrices $\mathbf{R}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$, $\mathbf{R}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT1}}$, $\mathbf{R}_{\mathrm{RT0}\leftrightsquigarrow\mathrm{AP}}$, and $\mathbf{R}_{\mathrm{RT1}\leftrightsquigarrow\mathrm{AP}}$ of size $M \times N$
output: d; select RT0 if $d = 0$, and RT1 otherwise.
`// select RT0 at initialization`
$d \leftarrow 0$;
for $frame \leftarrow 0$ **to** ∞ **do**
  calculate $\overline{\mathrm{SNR}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$, $\overline{\mathrm{SNR}}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT1}}$, $\overline{\mathrm{SNR}}_{\mathrm{RT0}\leftrightsquigarrow\mathrm{AP}}$ and $\overline{\mathrm{SNR}}_{\mathrm{RT1}\leftrightsquigarrow\mathrm{AP}}$ from Eq. (2);
  calculate Δ_{AP} and Δ_{UD} from Eq. (1);
  set d based on Table 1;
  assign $(N-1)$ data-slots to RTd and 1 data-slot to the other RT of the UD;
end

Table 1: Decision table for assigning RTs of the UD based on average SNR differences.

	if $\Delta_{UD} > T_0$	if $-T_1 \leqslant \Delta_{UD} \leqslant T_0$	if $\Delta_{UD} < -T_1$
if $\Delta_{AP} > T_0$	set $d \leftarrow 0$	set $d \leftarrow 0$	keep d
if $-T_1 \leqslant \Delta_{AP} \leqslant T_0$	set $d \leftarrow 0$	keep d	set $d \leftarrow 1$
if $\Delta_{AP} < -T_1$	keep d	set $d \leftarrow 1$	set $d \leftarrow 1$

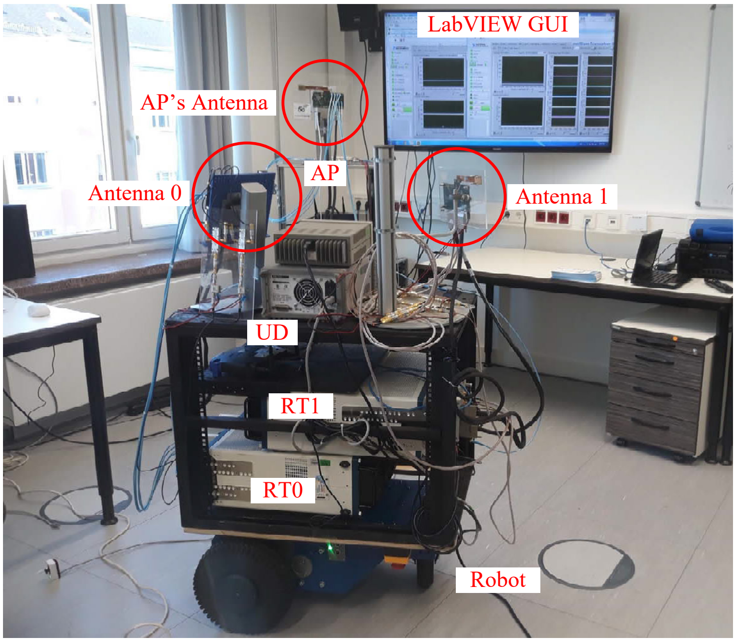

Figure 1: Hardware setup for the experiments.

Algorithm 1 shows how our method of data slot assignment to RT0/RT1 works. In this algorithm, $\mathbf{R}_{\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$ denotes a matrix of size $M \times N$ where its (i, j)th element is set to $\mathrm{SNR}_{i,j,\mathrm{AP}\leftrightsquigarrow\mathrm{RT0}}$ if slot j in ith previous frame is assigned to RT0 for transmission, and it is ignored otherwise. N denotes the total number of data slots in a frame. The other matrices are defined similarly. In the `for` loop, which is an infinite loop working as long as system is on, at each frame it first calculates Δ_{AP} and Δ_{UD} using Eqs. (1)–(3). Then using Table 1, it sets the value for d which is the output of this algorithm. The system assigns $(N-1)$ data-slots to RT0 and one data-slot to RT1 if $d = 0$ and conversely, $(N-1)$ data-slots to RT1 and one data-slot to RT0 if $d = 1$. We assign at least one data-slot to each RT of the UD for SNR calculations.

3.2. Solution Implementation

We implemented our proposed scheduling algorithm in the LabVIEW environment. In our testbed platform, the radio frame duration is 20 ms. We set the following values for the parameters in our experiments: $M = 20$, $N = 100$, $N_d = 100$, $N_s = 200$, $T_0 = 20$ dB, and $T_1 = 13$ dB. We obtained T_0 and T_1 empirically. They are different because the physical properties of antennas of RT0 and RT1 in our testbed are different and RT1 has a higher noise level.

4. Experimental Methodology

To evaluate the performance of our solution, we conducted two sets of controlled experiments called *movement* and *blockage* experiments. We have set a controlled environment for the experiments such that we could evaluate our hypothesis with a sufficient degree of confidence, i.e., with predictable movement, predictable blockage, etc. For instance, this experimental environment allowed us to systematically collect the measured throughput to show the benefit of employing the extra transceiver.

The goal of movement and blockage experiments is to show that when a device equipped with double RT loses the LoS of its first RT due to movement or blockage it can switch to its other RT and continue communication with the

AP. For this purpose, we use four metrics defined in the next section, namely stability, average response delay, average number of switchovers, and percentage of switchovers in short-time blockages.

We conducted all our experiments using a testbed for 60 GHz WLANs developed in MiWaveS project [8] shown in Fig. 1, which includes fully programmable PHY and MAC layers from National Instruments working at a bandwidth of 750 MHz using seven MCSs, and configurable phased arrays from SiBeam. The details of internal hardware setup of the UD and AP are available at [6, 8]. RT0 and RT1 are mounted on a robot which can move both devices. We use a rack to mount the hardware for these two devices. On top of this rack, we put the antennas of these RTs. This set represents one side of communication in our experiments. The other side is the hardware of AP behind this rack in the photo which is mounted on another rack. Its antenna is put on top of the rack to see the antennas of the UD. Behind it, the screen mounted on the wall shows the transmit and receive throughput of all devices implemented in LabVIEW.

For the movement experiments, we mount the antennas back-to-back and move the robot's front wheel to the right and left. In the right position, Antenna 0 can see the AP's antenna and in the left position, Antenna 1 can see it. In the left position, the AP communicates with the UD through RT1, Antenna 0 has no LoS with AP's antenna, and RT0's signal power is very weak such that its data rate is zero even with a strong MCS. When we move the robot to the right, Antenna 0 gets a good LoS with the AP's antenna and the communication goes through RT0.

For the blockage experiments, we adjust both UD's antennas to face toward AP's antenna. In this experiment, we simply block the LoS path of Antenna 0 or Antenna 1 to force the system to switch to the other RT for communication.

5. Experimental Results

To cover as many types of experiments as possible, we use seven MCSs and test three values for α in Eq. (2): $\alpha = 0.001$ for using only the "Last Frame SNR", $\alpha = 0.8333$ as a sample of "Weighted Averaging", $\alpha = 1$ for "Equal-Weight Averaging". This sums up to a total of 42 experiments for both movement and blockage setups. We also conducted the same experiments using the original setup without using our double-link implementation over seven MCSs for comparison. We call this "Single-RT" in this paper which is the setup used in [5]. This adds 14 more experiments summing the total number of experiments to 56. The duration of each experiment is one minute, which was determined empirically to provide sufficient time for reliable data capture.

5.1. Blockage Experiment

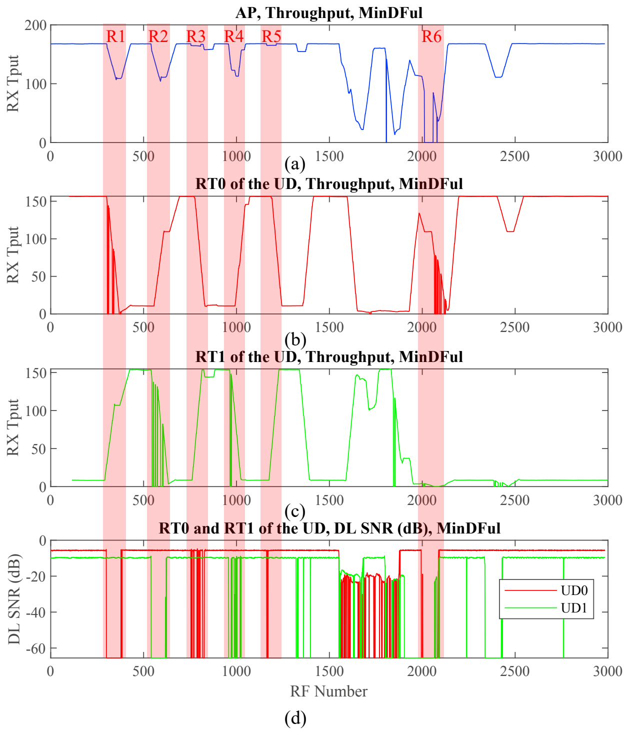

Figure 2: Received throughput and SNR values over time in MinDFul for blockage experiment using Last Frame SNR and MCS: 1/2 QPSK.

Figures 2-(a)-(c) show the received throughput in AP, and RT0 and RT1 of the UD over time, respectively. Figure 2-(d) shows the SNR values of DL side measured by RT0 and RT1. We used "Last Frame SNR" and set MCS to 1/2 QPSK. Here, the horizontal axis depicts frame numbers which are equivalent to time. As each frame size is 20 ms, 3000 frames are equivalent to 60 seconds, the total time in each experiment we captured data from the testbed. Times at which an event is occurred are highlighted with red bars named R1 - R6.

In Fig. 2-(d), a severe reduction in SNR from about -5dB to about -65dB indicates blocking of the LoS path. At first, all the traffic is carried through RT0, which is about 168 Mbps in the AP side and 156 Mbps in the RT0 side. The throughput of RT1 is about 8.5 Mbps. In region R1, we block antenna of RT0 causing the system to switch to RT1 to carry all traffic through RT1. The throughput of RT1 goes beyond 156 Mbps, while the throughput of RT1 decreases to almost zero. This switching causes a small reduction of AP's received throughput because of the delay in switching.

In R2, we block RT1's antenna causing the system to switch back to RT0. In R3 and R4, we block the LoS several times in front of RT0 and RT1's antennas, respectively. In both cases, the system switches to the other device. In R5, we block RT0's antenna in a small fraction of time by passing hand very quickly. As depicted, even with a quick short time blocking of the antenna the system switches to the other device. This is because here we use only the last frame's SNR for deciding on switching to the other device, and for a short period of time, the system experiences a severe drop in SNR and decides to switch to the other device immediately. In R6 we block both UD's antennas which resulted in AP's throughput dropping to 0 bps. After releasing the blockage in this region, the system returns to communication via RT0 which was the device it was using before this simultaneous blockage.

We observed that using SNR averaging methods explained in Eq. (2) lead to slightly different results. For example, in a similar experiment we used "Weighted Averaging" with $\alpha = 0.8333$ and blocked the active RT's LoS path quickly similar to R5 in Fig. 2. But, the system did not switch to the other RT. This is because the system averages the measured SNR over the last 20 frames, so a sudden drop in SNR does not result is a sudden drop in average SNR, and the system does not switch to the other RT of the UD.

To compare three SNR averaging methods, we measured some metrics for experiments similar to Fig. 2. The average response delay measured for switchovers from one RT to the other one from the moment we block the antenna to the time where switching takes place. In all 21 blockage experiments of MinDFul setup, we found 81 switchovers and measured this delay in each and averaged over them. These were 56.2, 70.0, and 74 ms for Last Frame SNR, Weighted Averaging, and Equal-Weight Averaging, respectively. As expected, Last Frame SNR is the fastest switching method, since it only uses the current frame's SNR to decide for switching, and when the SNR drops below the SNR of the other device plus a threshold, it immediately decides to switch RT. But, in the other two averaging methods, the AP uses the SNR values of all last 20 frames. There is no meaningful difference between the average delay of Weighted Averaging and Equal-Weight Averaging methods, while the second one shows a slightly more delay in responding which is because it uses equal weights for the SNRs of all previous frames, while the first one applies higher weights to the SNRs of most recent frames with respect to the older ones.

As another way to compare these three SNR averaging methods, we counted number of times a switchover occurred in one minute in blockage experiments for MinDFul setup in all 21 experiments. On average, Last Frame SNR, Weighted Averaging, and Equal-Weight Averaging made 9.4, 6.6 and 7.2 switchovers, respectively. As expected, Last Frame SNR made more switchovers because even in short-time blockages it switches RTs. Again, there is no meaningful difference between Weighted Averaging and Equal-Weight Averaging.

In all our 21 blockage experiments for MinDFul setup, we had 15 one-time short-time blockages and 35 multiple consecutive short-time blockages summing to a total of 50 such events. In 96% of such events with Last Frame SNR setting, the system switched to the other RT as expected. In Weighted Averaging and Equal-Weight Averaging, these percentages were 79% and 80%, respectively.

In summary, there is no significant difference between Weighted Averaging and Equal-Weight Averaging, but using Last Frame SNR results in faster response to events. This makes the system more sensitive to even short-time blockages which results in unnecessary number of switchovers.

5.2. *Movement Experiment*

Figures 3-(a)–(d) show received throughput and SNR values over time in MinDFul for the movement setup. Here we used Weighted Averaging with $\alpha = 0.8333$ and set MCS to 3/4 QPSK. We moved the UD to right and left three times starting from left position. In Fig. 3-(a)–(d), at first all traffic is going through RT1. In region R1, the SNR of RT1 drops to -65 dB and the SNR of RT0 raises to -5dB because of moving the UD from left to right by which making RT1's antenna's rear face to AP's antenna and lose its LoS with it and at the same time, making RT0's antenna front side face to AP's antenna and find a LoS with it. This causes the system to switch from RT1 to RT0 which transmits all traffic through RT0. For a short period of time in R1, both RT0 and RT1 experience -65dB in their SNR as both UD's antennas have an almost 90° angle with AP's antenna where there is no antenna coverage. If we use three antennas, we can cover all 360°. In R2, we move the UD from right to left where the system switches back to RT1. This continues until the end of Experiment's time which was 1 minute. Looking at Figs. 3-(b) and (c), we notice that each of RTs were used almost in 50% of time making AP able to double its channel utilization as one side of this communication.

To compare the results, we conducted the same experiment with the Single-RT setup. This is depicted in Fig. 3-(e). Comparing it with Fig. 3-(a) we notice how using two antennas can make the mmWave channel more stable and increases the channel utilization. In Fig. 3-(e) where we used one RT, the channel utilization is less than 50% as we could not use the channel when there was no LoS with the AP's antenna, while in Fig. 3-(a) we could achieve a utilization above 80% by switching to the second RT in such situations.

To measure how much stability of channel can be achieved with MinDFul, we define X%-Stability as the ratio of time where the received throughput was more than X% of the maximum achievable receive throughput. Here we use 70%-Stability in our plots. This way, for example if in an experiment the system can achieve more than 70% of maximum possible receive throughput in 90% of time, its 70%-Stability is 90%.

Figure 4 shows the 70%-Stability of RT0+RT1 received throughputs for MinDFul setup compared to Single-RT setup at different MCSs averaged over different SNR averaging methods in movement experiments. It is evident that with almost all MCSs, MinDFul shows superior stability compared to the Single-RT setup. This is our main achievement in these sets of experiments to show that using multiple RTs for communication can make the mmWave channel more stable and robust.

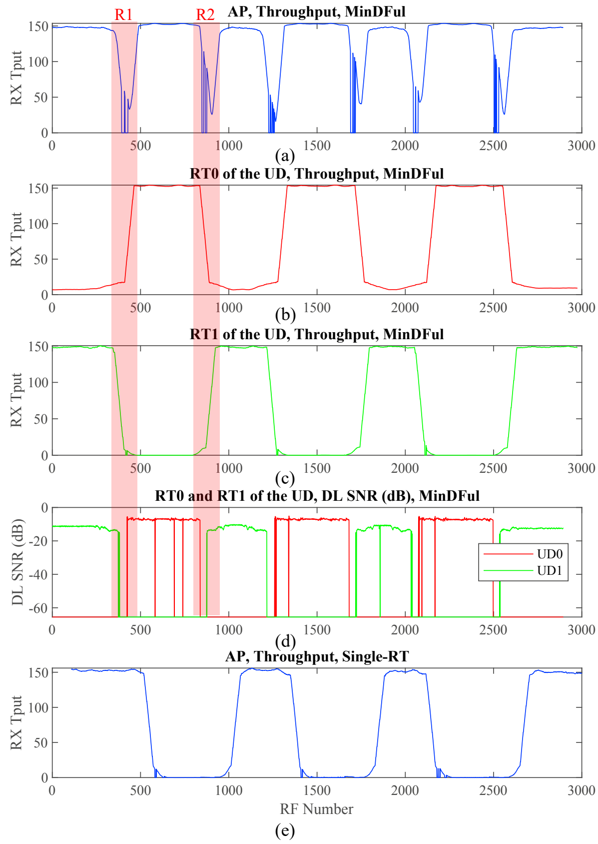

Figure 3: (a)-(c) Received throughputs and (d) SNR values over time in MinDFul for movement experiment using Weighted Averaging; (e) received throughput of AP in Single-RT setup; MCS: 3/4 QPSK.

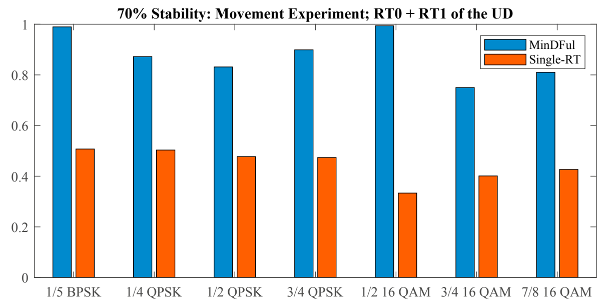

Figure 4: Average stability of MinDFul vs Single-RT setups for movement experiments.

6. Related Work

In recent years, with the advent of some mmWave laboratory testbeds, a considerable number of experiments have been conducted and reported in the literature. This involves measuring channel properties such as received power, attenuation and channel impulse response [7]. Using new testbeds with physical layer implementations including multiple MCSs, array antennas with beam steering protocols, software defined radios and baseband controllers, new experiments at the PHY layer have been developed, including parameters such as bitrate, beam steering algorithm accuracy and speed, array antenna beam pattern, error rates, delay and latency [9]. Recently, some testbeds have been equipped with MAC layer and some with end-to-end application layer functionalities which provide access to the full-stack network layers. In these cases researchers reported measurements of some MAC layer parameters including throughput, frame error rate, and MAC protocols performance. Some parameters are used as variables, including frequency, distance, antenna array size, MCS, reflecting/obstructing material, and Tx power [10]. The testbed developed in MiWaveS project is one of the most complete, providing measurements at all network layers [5] for which a more advanced version was recently developed [11]. The main contribution of MinDFul, which has not been explored to date, is using and validating double mmWave links as a practical solution.

Using multiple-input and multiple-output (MIMO) antennas can offer a faster approach for solving the blockage problem [4]. However, based on our own observations and other researchers [5], in the event of blocking LoS of the antennas, the existence of a non-LoS path cannot be relied upon because most materials do not show enough reflection properties for mmWave frequencies. Also, MIMO does not solve the problem of mmWave antennas having only 180° coverage. MinDFul is superior from this perspective.

7. Conclusion

mmWave links are vulnerable to movement and blockages. To make the mmWave channels more stable and robust against these events, we propose to use two sets of radio-frequency devices for each side of communication and switch from one set to the other one in response to a severe drop in channel quality. We designed and developed this idea as a scheduling algorithm on a mmWave testbed and conducted two sets of experiments to evaluate its performance in blockage and movement scenarios. The experimental results show that the main idea of using double links works and it can make the channel more stable. The channel utilization is almost doubled in the movement scenario, compared to using a single transceiver. We implemented two averaging methods over the measured SNRs of recent radio frames. Our results show using only the last radio frame's SNR leads to ∼15 ms less delay but makes it more sensitive to short-time blockages where the number of switchovers from one device to the other one increases.

For future work, the authors aim at making more field tests in particular in outdoor environments and carrying real AR traffic through their testbed with MinDFul.

Acknowledgements

This project received funding from the European Unions Horizon2020 research and innovation programme under the ORCA Project grant agreement No 732174 and under the Marie Skłodowska-Curie grant agreement No 713567. This publication has also emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number 13/RC/2077.

The authors would like to thank Clemens Felber from National Instruments for the technical support with respect to the LabVIEW software and hardware, and also Zhitao Lin from the Technische Universitt Dresden who maintained the testbeds remote access and network for the experiments. Both of them were also involved in the experiments through the H2020 ORCA project.

References

[1] Abari, O., Bharadia, D., Duffield, A., Katabi, D., 2017. Enabling high-quality untethered virtual reality, in: Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, USA. p. 531544.

[2] Ajorloo, H., Manzuri-Shalmani, M.T., 2016. Throughput modeling of distributed reservation protocol. IEEE Transactions on Mobile Computing 15, 503–515. doi:10.1109/TMC.2015.2417881.

[3] Dmitriev, S., 2019. Autonomous cars will generate more than 300 TB of data per year. URL: https://www.tuxera.com/blog/autonomous-cars-300-tb-of-data-per-year/.

[4] Ghasempour, Y., Haider, M.K., Cordeiro, C., Knightly, E.W., 2019. Multi-user multi-stream mmwave wlans with efficient path discovery and beam steering. IEEE Journal on Selected Areas in Communications 37, 2744–2758. doi:10.1109/JSAC.2019.2947820.

[5] Kadur, T., Chiang, H., Fettweis, G., 2018. Experimental validation of robust beam tracking in a nlos indoor environment, in: 2018 25th International Conference on Telecommunications (ICT), pp. 644–648. doi:10.1109/ICT.2018.8464878.

[6] Kadur, T., Chiang, H., Ohlmer, E., Felber, C., Ullmann, M., Schirok, J., Loehning, M., Swist, D., Dianati, M., Mi, D., Leinonen, T., Tamir, N., Potelon, T., Aurinsalo, J., Sily, J., Dehos, C., Marnat, L., , Mayrargue, S., 2017. Integrated Analog Front-Ends, Protocol and Software Infrastructure and Beamforming Implementation. Technical Report Deliverable D5.5. MiWaveS.

[7] MacCartney, G.R., Deng, S., Sun, S., Rappaport, T.S., 2016. Millimeter-wave human blockage at 73 ghz with a simple double knife-edge diffraction model and extension for directional antennas, in: 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1–6. doi:10.1109/VTCFall.2016.7881087.

[8] Medela, A., Chiang, H.L., Kadur, T., Putkonen, J., Z. Du, L.M., Dehos, C., Mayrargue, S., Ohlmer, E., Felber, C., Ullmann, M., , Swist, D., 2017. mmWave Access and Backhaul Link Tests and Presentation of Final Demonstrator. Technical Report Deliverable D6.5. MiWaveS.

[9] Nitsche, T., Flores, A.B., Knightly, E.W., Widmer, J., 2015. Steering with eyes closed: Mm-wave beam steering without in-band measurement, in: 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 2416–2424. doi:10.1109/INFOCOM.2015.7218630.

[10] Quadri, A., Zeng, H., Hou, Y.T., 2019. A real-time mmwave communication testbed with phase noise cancellation, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 455–460. doi:10.1109/INFCOMW.2019.8845251.

[11] Saha, S.K., Ghasempour, Y., Haider, M.K., Siddiqui, T., De Melo, P., Somanchi, N., Zakrajsek, L., Singh, A., Shyamsunder, R., Torres, O., et al., 2019. X60: A programmable testbed for wideband 60 ghz wlans with phased arrays. Computer Communications 133, 77–88. doi:10.1016/j.comcom.2018.09.005.