Masters' Thesis

Study on 60 GHz Wireless Networks for Virtual Reality

(バーチャルリアリティ向き 60 ギガヘルツ帯無線ネットワークの研究)

Ben Ruktantichoke
48-166472

The University of Tokyo
Information Science and Technology
Information and Communication Engineering
Advising Professor: Kaoru Sezaki

Abstract

Virtual reality (VR) playgrounds are augmented spaces where multiple users are able to physically interact simultaneously in real and virtual worlds. In such environments, untethered VR headsets are necessary for immersive experience. VR head mounted displays require high data rates that cannot be supported by Wi-Fi. 60 GHz signals are essential for enabling the required bandwidths. However, microwave signals suffer from signal attenuation and obstacle blockages. Relay VR networks overcome these issues by rerouting and amplifying signals in VR playgrounds.

Users in relay VR networks naturally form bipartite graphs that can be matched to determine a routing scheme. Relay VR networks change much more rapidly due to user mobility than typical networks, and frequent rerouting is needed to maintain user connectivity. The delays incurred from switching relays prevent existing matching algorithms from meeting latency requirements. These algorithms also repeatedly disconnect the same users over time when blockage is severe. Matching algorithms that prolong network stability and provide fair connectivity are crucial for relay VR networks.

Stable and fair matching algorithms are conceived by preserving user pairings over time and prioritizing the most frequently disconnected user in matching. Rerouting frequency is reduced and connectivity fairness is improved in all modified matching algorithms. In VR playgrounds, users and their movements contain structure that can be exploited in matching. A novel group matching algorithm is proposed for stable and fair bipartite matching in relay VR networks. Group matching outperforms all existing algorithms in stabilizing and ensuring quality of service in relay VR networks. Experiments also show that AP height significantly influences rerouting frequencies and matching failures. The results indicate that users' quality of experience in VR playgrounds can be guaranteed with relay VR networks.

Preface

これはゲーム、

そう思っていたー

Table of Contents

List of Figures

Maximum Connectivity in Static Relay Virtual Reality Networks

Prolonging Network Stability in Dynamic Relay Virtual Reality Networks

Guaranteeing Fair Connectivity in Dynamic Relay Virtual Reality Networks

List of Tables

Background

Prolonging Network Stability in Dynamic Relay Virtual Reality Networks

Guaranteeing Fair Connectivity in Dynamic Relay Virtual Reality Networks

List of Algorithms

Maximum Connectivity in Static Relay Virtual Reality Networks

Prolonging Network Stability in Dynamic Relay Virtual Reality Networks

Guaranteeing Fair Connectivity in Dynamic Relay Virtual Reality Networks

Abbreviations

3D	3-Dimensional
AMD	Advanced Micro Devices
AP	Access Point
FCC	Federal Communications Commission
FOV	Field of View
HD	High Definition
HMD	Head Mounted Display
LAN	Local Area Network
LOS	Line of Sight
MIMO	Multiple Input Multiple Output
mmWave	Millimeter Wave
P2P	Peer to Peer
PC	Personal Computer
PHY	Physical Layer
QoE	Quality of Experience
QoS	Quality of Service
RGB	Red, Green, Blue
SNR	Signal to Noise Ratio
VR	Virtual Reality

1 Introduction

Virtual reality (VR) is a form of entertainment that immerses users in surreal out-of-world experiences. The first generation of VR head mounted displays (HMDs), e.g. HTC Vive [12], Oculus Rift [20], are tethered to PCs that computes and renders graphics for the headset. Recent commercial solutions for untethering VRHMDs, such as TP Cast 2 [25], advertises using artifact-free compression that bounds the latency to 1ms. Modern VR systems are deployed as single-user platforms, but it is *not* difficult to imagine that multiuser VR attractions will soon be offered in amusement theme parks and gaming arcades. For this to happen, simultaneous wireless data transmission is necessary for every user in the VR space.

When multiple users are collocated in a VR space or playground, a VR wireless network is formed by the untethered headsets and access point(s) (AP). The HMDs require extremely high data rates even when data is compressed. In the case of the HTC Vive, its display resolution is 2160×1200 and refreshes at 90 Hz [12]. Assuming that 3 bytes are used to represent RGB values, the required data rate amounts to 5.59872 Gbps, which cannot be supported by traditional wireless technologies, i.e. Wi-Fi [28]. TP Cast 2 compresses the rendered frames by 50x [25], reducing the required data rate to 111.9744 Mbps, but still cannot be supported by Wi-Fi. Higher frequency signals are essential for VR networks.

Millimeter wave (mmWave) technology can be used to transmit multi-Gbps of data required by VR headsets, however microwave signals in the 60 GHz band suffer from obstacle blockage and signal attenuation. Relaying is known to improve system performance and coverage in general mobile networks and can be used to overcome the shortcomings of mmWave signals in VR networks. In mmWave networks, user nodes naturally form bipartite graphs consisting of nodes with adequate line of sight (LOS) connection to the AP, and nodes whose LOS connection is too weak or has been blocked. Users' quality of experience (QoE) can be improved by having the prior reroute and amplify signals for the latter. A routing scheme that minimizes network latency and overhead can be determined by matching the bipartite graphs. User connectivity is guaranteed in the case of perfect matches.

User connectivity is maximized when the number of matches is maximized. Maximal matching with Ford-Fulkerson algorithm does this in O(Ef) time, where E is the number of edges and f is the maximum flow in the graph [8], whereas the fastest maximal matching algorithm for n-vertex bipartite planar graphs has $O(n \log^3 n)$ time complexity [5]. Since VR networks are small, time complexity is not an issue. Section 3 evaluates maximal matching in relay VR networks.

Two important issues in relay VR networks are network stability and connectivity fairness. Virtual reality networks change much more rapidly due to user mobility than mobile networks. In multi-user scenarios, mmWave wireless links are unreliable. Even when there is no obstacle blockage, users themselves can block communication links. LOS communication links change continuously over time, and existing relay schemes undergo frequent rerouting to maintain user connectivity. VRHMDs require steady streams of packets, and frequent rerouting will degrade users' QoE. Experiments show that latency requirements cannot be met because of switching delays. Relay VR networks need relay schemes that are stable over time.

In section 4, a number of matching algorithms are devised to maximize network stability over time. The matching algorithms prioritize previously established links and relies on nodes' rerouting frequency for prolonging network stability. The algorithms include a baseline greedy algorithm, modified versions of maximal and stable matching [17], and a hierarchical group matching algorithm. Since multi-user VR applications are typically goal-oriented and team-based, users' behavior and movements follow some structure. Group matching exploits this structure and produces the most stable routing decisions.

The other issue is matching failures occur for some node configurations even when maximal matching is used. These matching failures persist over time and existing algorithms tend to fail to match the same particular nodes. When a node repeatedly experience disconnect while other nodes do not, the quality of service (QoS) is unfair for that node. Fair connectivity is obtained by equally distributing this disconnect to all nodes. This means all nodes experience a brief moment of disconnect, as opposed to few nodes experiencing regular lag in gameplay. In section 5, the modified matching algorithms are optimized for connectivity fairness. Nodes that have experienced the greatest amount of disconnect are prioritized in greedy and maximal matching. These nodes receive preferential priority in stable and group matching. The modified algorithms guarantee fair connectivity in relay VR networks.

The numerical results demonstrate that the proposed fair matching algorithms maintain network stability while providing fair QoE in relay VR networks. Furthermore, AP height significantly affects the degree in which blockage occurs in VR playgrounds. Fair group matching makes relaying an ideal solution for enabling VR networks.

2 Background

This chapter serves to outline the motivating ideologies and different technological aspects of relay VR networks. Section 2.1 introduces motivations for developing relay VR networks. Section 2.2 defines VR playgrounds. Section 2.3 provides background on mmWave technology. Section 2.4 describes the simulation model, and section 2.5 summarizes related works.

2.1 Motivation

Relaying is the solution for using mmWave technology with untethered VRHMDs in VR playgrounds. Unlike mobile user equipment, VR headsets are worn on people's heads and are naturally in vantage points suitable for LOS communication. mmWave antenna arrays are generally placed atop the headset to facilitate omnidirectional communication. These arrays are rarely blocked by user themselves, and the probability that multiple users are simultaneously blocked is low. The worst case scenarios are actively avoided because users react to any decline in QoE by moving to better positions. Relaying should guarantee connectivity in all practical scenarios. Section 3 shows significant gains in QoE when relaying is allowed in VR networks.

Relay VR networks scale with user density to a certain degree. The number of relays increases with the number of users because each user has relaying capacity. As the number of relays increase, network coverage improves because there are more ways to route signals around obstacle blockages. In highly congested playgrounds, most users cannot act as relays because they are blocked and increasing the number of users would exacerbate the situation. It is hard to imagine that such a scenario will take place. Multi-user VR games usually do not require users to be so closely packed that would lead to relaying difficulties.

Relay VR networks are easy to deploy because only mmWave transceivers and microprocessors, such as AMD's Accelerated Processing Unit [11], need to be integrated with VRHMDs. No extra devices other than the VR system itself and a single mmWave AP are needed for a VR playground. The cost of these hardware components is amortized by reduced deployment costs and complexity. As for networking, relaying schemes maximizing throughput and minimizing delay are found with ease using matching algorithms. Sections 4 and 5 show that the two major issues with relay VR networks, relaying stability and fairness, are solved by adding simple heuristics to the matching algorithms. These factors make relaying befitting for VR networks.

2.2 Virtual Reality Playgrounds

VR playgrounds will revolutionize theme parks and arcades as attractions providing surreal experiences. The playgrounds are envisioned to be provided as a commercial service. It is important for services to maintain consistent QoS. In other words, it is acceptable to sacrifice QoE for few users if the overall service is improved.

Multiplayer VR applications can be enabled by virtually connecting users over the Internet. However, VR experience is not entirely immersive as users control their virtual avatar with a console and are not physically engaged. The virtual environment inside a VR playground is an augmented reality where users are able to interact simultaneously in virtual and real worlds.

In this work, multiplayer wireless VR playgrounds are simulated and relay VR networks are studied. It is difficult to obtain user traces from actual VR playgrounds, so a simulation model is constructed as a virtual environment with spatial constraints. VR playgrounds mimic field and courts in indoor and outdoor sports. The AP is placed in a location which is practical both scenarios. Physical objects and obstacles such as tables, walls, pillars, and nets can be simulated but are omitted for simplicity. This simplification does not detract from the results because experiment parameters have been tuned to favor blockage. Relaying is capable of dealing with all types of blockages

In order to meet the computational and latency demands of VRHMDs, VR playgrounds require dedicated computing resources. These resources must be collocated with VR playgrounds to achieve minimum network latency. In future 5G networks, small cells will be equipped with computational resources as part of edge computing. VR playgrounds can utilize edge computing to render graphics and host the virtual environment. The performance of mobile computer vision applications with edge computing is provided in appendix E.

2.3 Millimeter Wave Technology

The 60 GHz millimeter wave band supports multi-gigabit wireless communication but at only limited distances due to signal propagation loss and obstacle blockage. Recent studies [1, 2, 23, 33] have shown that 60 GHz signals are still reliable at 100m indoors and outdoors and under various settings. Microwave signals are slightly more susceptible outdoors, but there are no significant differences under ordinary conditions. This makes them appropriate for VR playgrounds.

Since 60 GHz signals have short wavelengths, compact electronically steerable phased arrays can be integrated with VRHMDs. Planar patch elements can be packed densely to enable

high bandwidth and narrow directional beams that limit interference. These patch elements inherently have a coverage of less than half-space but can jointly cover the entire space with multiple patches.

mmWave APs can be equipped with multiple antenna arrays to transmit independently to each user. 802.11ad supports data transmission rates up to 8 Gbps and phase array antenna beamforming [19]. There are 6 channels with 2.16 GHz bandwidth each. 802.11ay bonds four of those channels together for a maximum bandwidth of 8.64 GHz. MIMO is added with a maximum of 4 streams. The link rate per stream is 44 Gbps [15]. This means the combined raw data rate of multiple VRHMDs can be supported.

mmWave radio uses highly directional beams that require transmitter and receiver beams to be aligned. User mobility causes beam misalignment which breaks the communication link. [1] solves the problem of beam alignment and tracking by utilizing tracking information available in VRHMDs. [13] provably finds the optimal antenna beam in logarithmic number of measurements, works within the existing 802.11ad standard for mmWave LAN, and can support both clients and access points.

In a dense playground, the antenna arrays must produce pencil-like beams in order to have near-perfect spatial reuse. Due to the FCC's regulation, narrower beams must have their power reduced which restricts transmission range [30, 33]. Interferences such as reflections and side lobes also limit spatial reuse [23, 24, 27, 33]. [24] identifies that the sensitivity of microwave signals can be harnessed to diagnose link outages, thereby facilitating judicious protocol reactions. [27] uses a single pair of 60 GHz transmitter and receiver to resolve dominant reflection paths and passes this information into a ray-tracer to predict channel and network performance of arbitrarily located links.

Recent studies improve mmWave networks by using pose information [26] and multiband chipsets [22]. Neither of these works benefit VR networks as VRHMDs do not suffer from limited FOV like mobile handsets, and switching to WiFi frequencies in the event of blockage does little to improve VR application quality. VR networks can only be improved significantly through relaying.

2.4 Simulation Design

2.4.1 Virtual Reality Playground Model

VR playgrounds are simulated as a 3-dimensional space discretized as points on a 2-dimensional field with varying height. There can only be one node at a point, and each point

represents an area of $1m^2$. Three playgrounds are simulated in sections 3, 4, and 5: a 10x10m, 20x20m, and 30x30m playground with up to 4, 16, and 32 users respectively. This limits the maximum user density to 1 user/$25m^2$. The AP is placed at 2m or 4m height in the middle of one side of the playground. Placing the AP on the ceiling in the middle of the room would provide larger field of view (FOV) for the AP and reduce blockage occurrence. In practical scenarios especially outdoors, it may not be possible to place the AP at such vantage points. The AP has complete global information and disseminates routing decisions to all users.

User are modeled as cylinders with 25cm radius and randomly distributed height between 120cm and 200cm. LOS is determined by vectors to and from users' heights in 3D coordinates (x, y, height). Blockage occurs when the vector intersects another user's cylinder. Self-blockage is excluded as antenna arrays are placed atop the VRHMD where the probability of self-blockage is low. All other obstacles are omitted from the simulated space. Since receive and transmit channels are typically on opposite sides of the headset, interference is assumed to be negligible. Communication occurs only on LOS.

2.4.2 Millimeter Wave Channel Model

60 GHz links and their channel capacities are modeled using link budget models and Shannon's capacity respectively. The derivation follows [30]. The link budget model according to Friis free-space path loss formula in dB scale is,

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - PL \tag{1}$$

where P_{TX} is the transmit power, P_{RX_r} is the received power at distance d, and G_{TX} and G_{RX} are the antenna gains for transmit and receive antennas correspondingly. The FCC limits the equivalent isotropic radiated power in the 60 GHz band to a maximum power density of 9 $\mu W/cm^2$ at 3 meters from the radiating source [30]. If 40 dBm is emitted, the antennas must have 0 dBi gain. The received signal strength is attenuated by oxygen absorption and is dominated distance. Path loss can be expressed as,

$$PL(d) = 10\log_{10}\left(\frac{4\pi d}{\lambda}\right)^n + \xi, \qquad \xi \sim (0, \sigma^2) \tag{2}$$

where λ is the wavelength corresponding to carrier frequency f, n is the path loss exponent, and σ^2 is the lognormal shadowing variance. The minimum received power should be greater than noise levels at the receiver to ensure adequate performance. The required sensitivity at the receiver is,

$$P_{RX} \geq NF + F + SNR \tag{3}$$

where NF is the noise floor, F is the noise figure, and SNR is the signal to noise ratio at the receiver. When thermal noise is the primary source of interference, the noise floor is

$$NF = kTWF \tag{4}$$

where k is Boltzmann's constant, and T is the room temperature. In indoor scenarios, the noise floor is calculated to be -76 dBm. Assuming the noise figure is 0 dBm, the relationship between SNR and distance is,

$$SNR \leq 116 - 10\log_{10}\left(\frac{4\pi d}{\lambda}\right)^{n} + \xi \tag{5}$$

Using the Shannon's theorem, the relationship between channel capacity and communication distance is derived as,

$$C \leq W\log_2\left(1 + 10^{(116-10\log_{10}\left(\frac{4\pi d}{\lambda}\right)^{n}+\xi)/10}\right) \tag{6}$$

where W is the system bandwidth. Since this is the ideal maximum channel capacity, the required data rate is selected to be the uncompressed raw data rate of 5.59872 Gbps.

System Bandwidth	2.16 GHz
Carrier Frequency	61.56 GHz
Lognormal Fading Variance	5.8 dB
Path Loss Exponent	2

Table 1. Channel Simulation Parameters

2.4.3 Latency Model

HTC Vive operates at 90 Hz, so there can be at most 11ms of delay between frames. [32] provides latency values for a commodity VR system when 802.11ad is used to transmit frames. It takes 6.1ms to render graphics and 5.9ms to send and receive compressed frames. When there is no data compression, the transmission delay can be calculated from the previous section by dividing the required data rate by the channel capacity. According to [32], network processing delay is calculated to be 2.4ms by subtracting the send and receive delay by the time required to transmit 95% quality JPEG compressed frames over an 851 Mbps link. Relaying doubles the required data rate and transmission delay. The relay processing delay is shown to be negligible in [11] which demonstrates 50 Gbps forwarding performance.

mmWave communication incurs a delay involving beam alignment. The beam alignment delay for an antenna array of 256 elements for one client is 1.01ms [13]. More delay is incurred when the same array needs to align to multiple clients, however separate arrays can be used to

transmit to each client. The beam alignment delay is assumed to be identical for P2P links. The total delay for displaying a frame over an established link is equal to rendering delay (6.1ms) + network delay (2.4ms) + beam alignment delay (1.01ms) + transmission delay (5.59872/*C*s). In the case of relaying, the transmission delay is 11.19744/Cs.

When communication links change or break due to blockage, delays are incurred for beam training and recovery respectively. It is shown that link switching latency is at most 25ms in [22] and 295ms is taken on average to recover from blockage [26]. These delays are added in their respective scenarios.

Rendering	6.1 ms
Network	2.4 ms
Beam Alignment	1.01 ms
Transmission	$5.59872/C \times 1000$ ms
Relaying Transmission	$11.19744/C \times 1000$ ms
Switching	25 ms
Blocked	295 ms

Table 2. Latency Simulation Parameters

2.5 Related Works

VR Applications VR applications range from ultra wide panoramic video streaming in mobile networks to spatially interactive multiplayer games in VR playgrounds. Recent studies explore how 360° VR video streaming should be enabled in 5G networks [16, 18]. The results from these studies are inapplicable to VR playgrounds because multiplayer VR games have much more stringent requirements compared to 360° VR video streaming. These requirements include: each user must receive unique visual and audio content based on their location and head orientation, user inputs must be handled and displayed in a manner which is consistent with the virtual world, and the associated processing and transmission delay must not exceed 11ms to maintain VR immersion [12].

VR Networks Recent studies on multi-user virtual reality wireless networks optimizes subcarrier allocation [6] and propose multi-path routing for wireless transmission [10]. These studies neglect LOS requirements imposed by mmWave signals and assume endpoint users can be reached directly. In multi-user scenarios, mmWave wireless links are unreliable, and relay schemes must account for this unreliability.

Relay mmWave Networks [1] implements untethered VR and shows that mmWave technology can be used to stream multi-Gbps of data required by a single VRHMD. They introduce programmable mmWave mirrors that reroute and amplify signals when blockages occur. Multiple mirrors can be set up to obtain better system guarantees, yet it would still be difficult to guarantee connectivity in multi-user scenarios. As user density increases, the exact placement of mirrors takes on a larger role in determining node connectivity. Installing many mirrors is time consuming and tedious, even more so outdoors because effective mirror locations are limited. Instead of relying on mirrors, multiple users can be supported by having the users themselves act as relays.

Distributed association and relaying with fairness in mmWave networks is discussed in [29]. They introduce centralized and distributed solutions for load balancing among multiple APs and user relays. The centralized approach is formulated as a variant multi-dimensional assignment problem with nonlinear objective, which can be relaxed to a non-linear convex optimization problem [29]. When there is only one AP, the optimization problem can be solved by maximally selecting the highest capacity links when forming node and relay pairs. Their proposed methods are fair in terms of load balancing, but relay VR networks need to be stable and provide connectivity fairly to all users over time.

[29] also states that a centralized coordinator for client association and relaying is hard or impossible to have in practice. On the contrary, a centralized coordinator is appropriate for relay VR networks because of three reasons. One, relay VR networks are provided with commercial VR playgrounds that are operated by businesses. Two, centralized solutions avoid distributed computations imposed on resource constrained user nodes, and three, it is convenient because multiplayer VR gaming platform must inherently retain complete global information in order to correctly render graphics for each user.

Relay Mobile Networks Recent works in relay mobile networks include [21], which optimizes user connectivity and average delay by allocating relay APs, and [31], which optimizes system performance with stable user pairings. If user nodes are allowed to act as relays in [21], their scheme is closely approximated by maximal matching for relay VR networks. On the other hand, [31] is exactly stable matching. Stable here refers to user pairings that are fair in the sense that no unmatched pair prefers each other over its current pairing. Both algorithms produce optimal routing schemes, but may not necessarily be stable over time or provide fair connectivity due to significance of user mobility in relay VR networks as opposed to relay mobile networks.

3 Maximum Connectivity in Static Relay Virtual Reality Networks

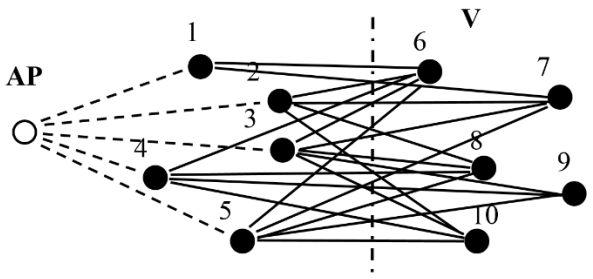
This section evaluates QoE of users in static VR networks and demonstrates benefits of relaying. The introduction provides rationale for relay VR networks. Bipartite matching is formulated in section 3.2. Section 3.3 formalizes baseline matching algorithms and proposes a novel group matching algorithm. Section 3.4 describes experimental design. Section 3.5 shows the numerical results, and section 3.6 discusses them.

3.1 Introduction

VR networks are ad-hoc networks that enable multi-user VR playgrounds. It is important for VR networked solutions to be minimalistic so they can be widely deployed with ease. A single mmWave AP does not guarantee users' QoE, and setting up multiple APs is laborious and requires technical expertise. Meanwhile, relaying can be easily integrated with VRHMDs. Experiments show that relaying significantly improves average data rate per user. Even when there are no blockages, relaying can improve QoE in static VR networks. Relay VR networks achieves maximum connectivity when blocked users are maximally matched with LOS relays.

3.2 Static Bipartite Matching Problem Formulation

Relay VR networks consist of n nodes and a single AP. The inputs are a set of vertices $\boldsymbol{W} = \{1, .., n\}$, a set of blocked nodes $\boldsymbol{V} \subset \boldsymbol{W}$, and a function $LOS(x, y)$ that returns a Boolean value indicating whether there is LOS between vertices x and y. LOS blockages are limited to only user blockages. The desired output is a set of node pairs $\boldsymbol{S} = \{(1, p_1), (2, p_2), \dots, (n, p_n)\}$, where $p_i \in \{1, \dots, n\}/\{i\}$ is i's parent that relays communication from the AP. If a node i has LOS to the AP and is communicating directly with the AP, then $p_i = \emptyset$. Maximum connectivity is obtained when $p_v \neq \emptyset$ for $\forall v \in \boldsymbol{V}$. In figure 1, the inputs are $\boldsymbol{W} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $\boldsymbol{V} = \{6, 7, 8, 9, 10\}$, and $LOS(x, y)$ is true for every edge shown.



Inputs:

$$W = \{1,2,3,4,5,6,7,8,9,10\}$$
$$V = \{6,7,8,9,10\}$$

LOS are the edges

Output:

$$S = \{(1,\emptyset),(2,\emptyset),(3,\emptyset),(4,\emptyset),(5,\emptyset),(6,1),(7,2),(8,3),(9,4),(10,5)\}$$

Figure 1. A bipartite graph consisting of user nodes in a VR network.

A possible perfect match is $\boldsymbol{S} = \{(1,\emptyset),(2,\emptyset),(3,\emptyset),(4,\emptyset),(5,\emptyset),(6,1),(7,2),(8,3),(9,4),(10,5)\}$. Perfect matches minimize network latency and overhead for resource-constrained user nodes while also reducing PHY layer interferences. If i does not have LOS to the AP and $p_i = \emptyset$, then i cannot be reached. The constraints of the problem are one-hop relaying, for $\forall p_i \neq \emptyset$, $p_m \neq i$ for $\forall m \in \boldsymbol{W}$, and relay nodes can only have one child node, $p_i \neq p_j$ for $\forall i,j \in \boldsymbol{W}, i \neq j$.

3.3 Matching Algorithms

In this section, possible baseline matching algorithms for relay VR networks are provided. Greedy matching serves as a naïve comparison baseline. Maximal and stable matching represent the optimizations performed om previous works pertaining to relay networks. A novel group matching algorithm that takes advantage of node groups is proposed.

3.3.1 Greedy Matching

Greedy matching performs a single pass over all blocked nodes and greedily pairs them with their most preferred parent.

ALGORITHM 1: Greedy Matching

1 : For each node v in$\boldsymbol{V}$ s.t. $p_v = \emptyset$:
2 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V}$ s.t. for $\forall u \in \boldsymbol{U_v}$: $LOS(v,u)$ and $LOS(u,AP)$; //Candidate Parents
3 : Sort $\boldsymbol{U_v}$ by v's preference; //Sort by link capacity
4 : For each node u in $\boldsymbol{U_v}$: //Match parent with highest capacity
5 : If $p_u = \emptyset$ and $p_i \neq u$ for $\forall i \in \boldsymbol{W}$:
6: $p_v = u$;
7: Break;

The list of candidate parents is sorted in line 3 by their link capacities as defined in section 2.4.2.

3.3.1 Maximal Matching

Maximal matching finds a matching with maximum number of pairs [17]. By selecting the closest parent for each node, the algorithm approximates the optimization performed in [29]'s centralized scheme. Ford-Fulkerson algorithm [8] is implemented using recursion in line 8 of algorithm 2, which is computationally acceptable for relay VR networks.

ALGORITHM 2: Maximal Matching

1 : For each node v in $\boldsymbol{V}$:
2 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V}$ s.t. for $\forall u \in \boldsymbol{U_v}$: $LOS(v,u)$ and $LOS(u,AP)$; // Candidate Parents
3 : Sort $\boldsymbol{U_v}$ by v's preference; //Sort by link capacity
4 : For each node v in $\boldsymbol{V}$ s.t. $p_v = \emptyset$:
5 : For each node u in $\boldsymbol{U_v}$: //Match parent with highest capacity
6 : If $p_u = \emptyset$ and $p_i \neq u$ for $\forall i \in \boldsymbol{W}$:
7 : $p_v = u$;
8 : Else if $\exists i \in \boldsymbol{V_t}$ s.t. $p_i = u$ and i can be assigned a new parent j:
9 : $p_i = j, p_v = u$;

As in greedy matching, the list of candidate parents for each node is sorted by their capacities in line 3 before matching is performed.

3.3.1 Stable Matching

Stable matching considers the preference of both blocked and relay nodes, and produces a matching where all pairs have no incentive to undermine their assignment by joint action [17]. "Stable" here refers to pairing stability. [31] optimizes system performance by ensuring pairing stability. Blocked nodes prefer relays with higher capacities while relays prefer relaying on higher capacity links. Parents have precedence over blocked nodes. This is implemented using Gale-Shapley algorithm [9].

ALGORITHM 3: Stable Matching

1 : For each node v in$\boldsymbol{V}$:
2 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V}$ s.t. for $\forall u \in \boldsymbol{U_v}$:
3 : $LOS(v,u)$ and $LOS(u,AP)$; //List of candidate parents
4 : Sort $\boldsymbol{U_v}$ according to v's preference; //Sorted by link capacity
5 : While $\exists v$ in $\boldsymbol{V}$ s.t. $p_v = \emptyset$ and $\exists q \in \boldsymbol{W} - \boldsymbol{V_t}$ s.t. $p_i \neq q$ for $\forall i \in \boldsymbol{W}$: /
6 : If $\exists u \in \boldsymbol{U_v}$ s.t. $p_u = \emptyset$ and $p_i \neq u$ for $\forall i \in \boldsymbol{W}$:
7 : $p_v = u$;
8: Else if $\exists i \in \boldsymbol{V}$ s.t. $p_i = u$:
9: If u prefers v over i: //Prefer nodes with higher capacity links
10: $p_v = u, p_i = \emptyset$;

The candidate parents for each node is sorted by channel capacity in line 4. Relay nodes' preference are considered in line 9.

3.3.1 Group Matching

Group matching takes advantage of node group distributions and prioritizes in-group relaying as a heuristic. Node groups are given as additional inputs to the problem. A node group is formalized as $\boldsymbol{G_i} \subseteq \boldsymbol{W}$ for $i \in \{1, \dots, m\}$, where m is the total number of groups, $\boldsymbol{G_i} \cap \boldsymbol{G_j} = \emptyset$ for $\forall i, j \in \{1, \dots, m\}, i \neq j$, and $\boldsymbol{G_1} \cup \boldsymbol{G_2} \cup \cdots \cup \boldsymbol{G_m} = \boldsymbol{W}$. Group matching performs stable matching separately among group members first, and then among non-group members. Parent nodes prefer group members over non-group members, and then use distance for tie-breaking.

ALGORITHM 4: Group Matching

1 : Let $v \in \boldsymbol{G_{v,j}}$ for $j \in \{1, \dots, m\}$; //$\boldsymbol{G_{v,j}}$ denotes v's group

2 : For each node v in $\boldsymbol{V}$: //Sort in-group and out-group parent sets

3 : Let $\boldsymbol{U_{g,v}} \subseteq \boldsymbol{G_{v,j}} - \boldsymbol{V}$ and $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V} - \boldsymbol{U_{g,v}}$ where for $\forall u \in \boldsymbol{U_{g,v}}$ and $\forall u \in \boldsymbol{U_v}$:

3 : $LOS(v,u)$ and $LOS(u,AP)$;

4 : Sort $\boldsymbol{U_{g,v}}$ and $\boldsymbol{U_v}$ according to v's preference; //Sort by link capacity

5 : While $\exists v$ in $\boldsymbol{V}$ s.t. $p_v = \emptyset$ and $\exists q \in \boldsymbol{W} - \boldsymbol{V}$ s.t. $p_i \neq q$ for $\forall i \in \boldsymbol{W}$:

6 : If $\exists u \in \boldsymbol{U_{g,v}}$ s.t. $p_u = \emptyset$ and $p_i \neq u$ for $\forall i \in \boldsymbol{W}$: //In-group matching

7 : $p_v = u$;

8 : Else if $\exists u \in \boldsymbol{U_{g,v}}$ s.t. $p_u = \emptyset$, and $\exists i$ s.t. $p_i = u$:

9 : If $i \in \boldsymbol{V} - \boldsymbol{G_{v,j}}$ or u prefers v over i: //Prefer nodes with higher capacity links

10: $p_i = \emptyset, p_v = u$;

11: Else if $\exists u \in \boldsymbol{U_v}$ s.t. $p_u = \emptyset$ and $p_i \neq u$ for $\forall i \in \boldsymbol{W}$: //Out-group matching

12: $p_v = u$;

13: Else if $\exists u \in \boldsymbol{U_v}$ s.t. $p_u = \emptyset$, and $\exists i \in \boldsymbol{V} - \boldsymbol{G_{u,j}}$ s.t. $p_i = u$:

14: If u prefers v over i: //Prefer nodes with higher capacity links

15: $p_v = u, p_i = \emptyset$;

The list of in-group and out-group candidate parents for each node is sorted in line 3. In-group pairs are made first in lines 6-10 followed by out-group pairs in lines 11-15.

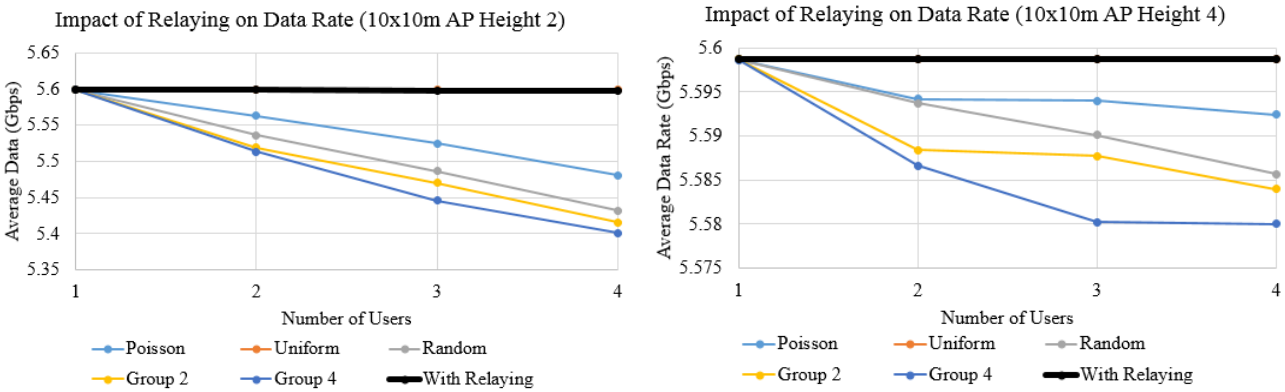
3.4 Static Experiment Design

Experiments are performed using three node distributions: uniform, Poisson, and group random. Uniform distributions represent ideal scenarios where users are evenly spaced apart. Users fill consecutive columns of a perfect square grid until a larger grid is needed. Only users' heights change across trials.

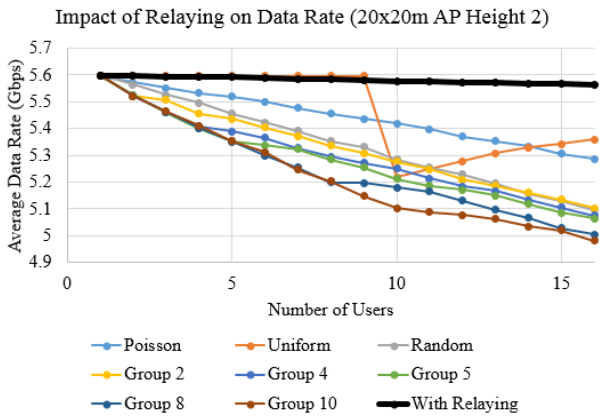
Poisson distributions are discussed in [3] and are generated as in [7]. The simulation space is divided into sub-regions, and a Poisson value is drawn for each sub-region to generate the respective number of nodes. Four poisson sub-regions are generated for VR playgrounds with four users or less while 16 poisson sub-regions are generated when there are more than 4 users. Poisson distributions tend to be clustered and represent VR applications with interaction among users.

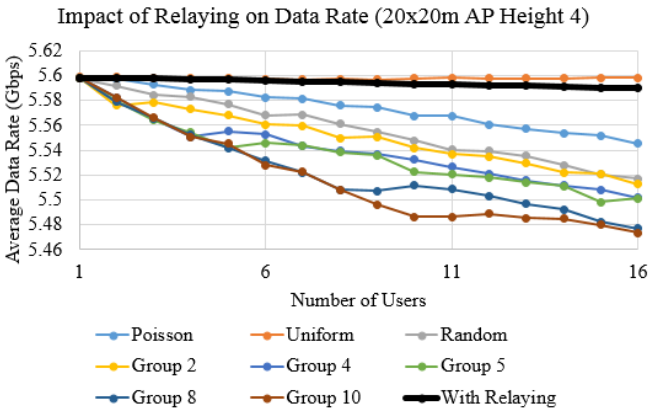
Group random distribution generates specific sizes of groups. The position of each group is determined by its leaders' position, which is randomly generated in the entire space. Group members are also generated randomly, but in a square space with the group leader at the center. The dimensions of the square space are 6x6m, 8x8m, and 10x10m for 10x10m, 20x20m, and 30x30m VR playgrounds respectively. The generated group sizes are 1, 2, 4, 5, 8, and 10. Random group distributions represent team-oriented games, each group is independent. 10,000 trials were performed in each experiment.

3.5 Numerical Results

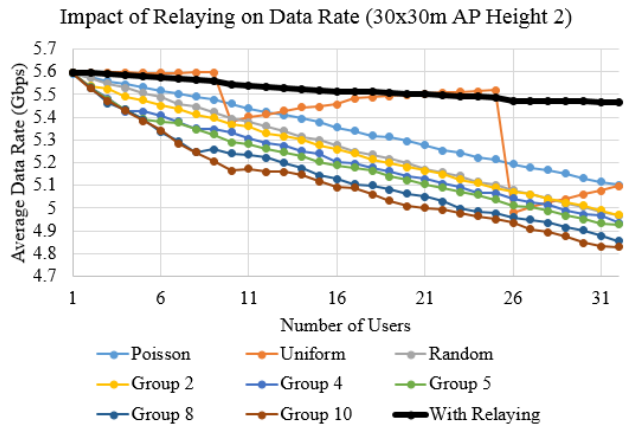


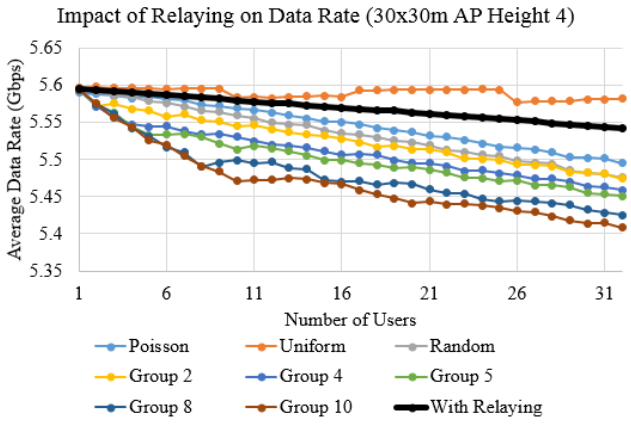
Figures 2 (left) and 3 (right). The average data rate per user when there is no relaying in 10x10m VR playgrounds is shown with respect to the number of users for each distribution. Maximal matching achieves the best data rates shown in bold. Relaying performance is averaged over all distributions. Figure 2 on the left and figure 3 on the right show the results when the AP is placed at 2m and 4m height respectively.





Figures 4 (left) and 5 (right). The average data rate per user in 20x20m VR playgrounds is shown with respect to the number of users for each distribution.





Figures 6 (left) and 7 (right). The average data rate per user in 30x30m VR playgrounds is shown with respect to the number of users for each distribution.

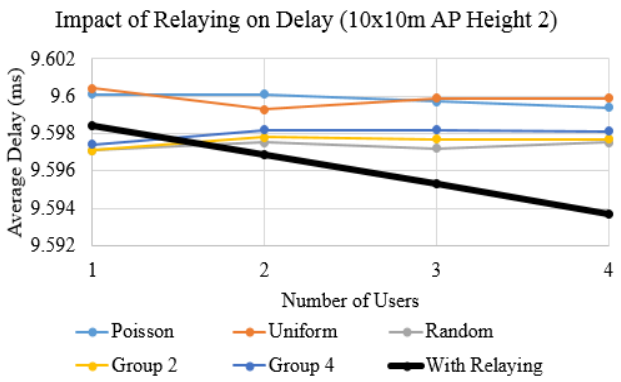
Relaying with maximal matching achieves the best data rates in all experiments. The maximum data rate is obtained when there is only one user because blockage does not occur. The average data rate with and without relaying decreases as the number of users increases because blockage becomes more severe. Blockage is also more severe when the AP is at 2m height. When more nodes are blocked, the probability of matching failure increases. If nodes cannot be matched, their data rates are zero, thereby reducing overall connectivity.

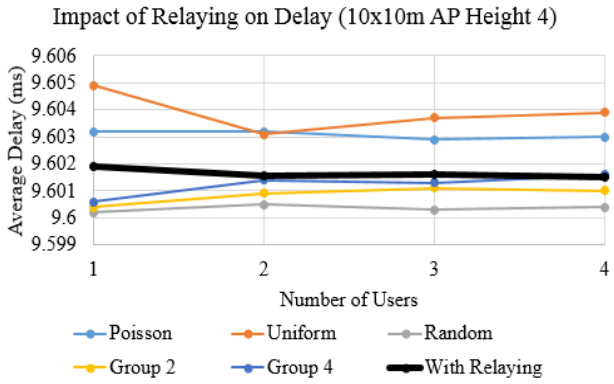
In general, uniform distributions contain the least amount of blockage, followed by Poisson and group random distributions. The average data rate for uniform distributions changes with the number of users in each experiment. Users are redistributed when a larger perfect square grid is allocated to accommodate increasing number of users. The exact positions of user nodes nearest to the AP determines how many nodes are blocked. When grid columns line up with

the AP, more nodes are blocked. The remaining grid positions likely have LOS to the AP. As nodes fill these positions, the average data rate typically improves.

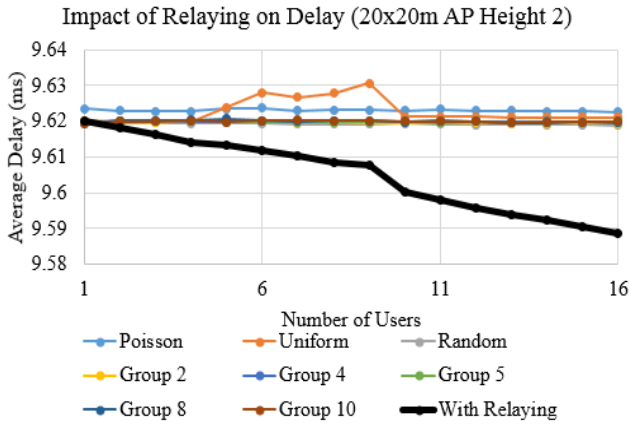
User nodes are typically more congested in Poisson distributions than random distributions, yet the average data rate is consistently higher for Poisson distributions. This indicates that greater congestion does not necessarily lead to more blockages. Experiments indicate that nodes in Poisson distributions tend to occupy smaller FOV angles as seen from the AP.

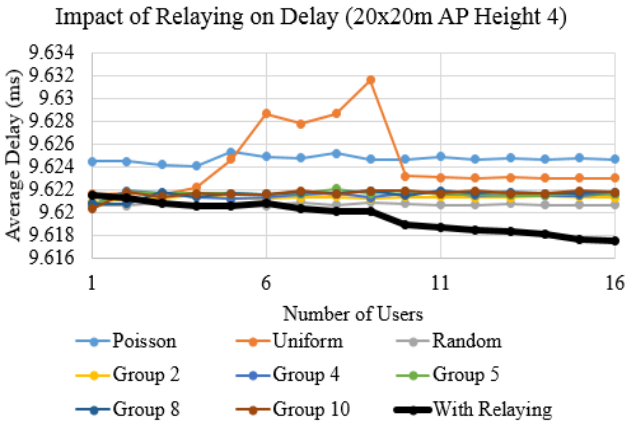
Blockage occurs more severely in group random distributions because nodes in the same group are packed into a square space. Group members occupy a larger portion of one another's FOV compared to non-group members. More blockages occur when group size increases.



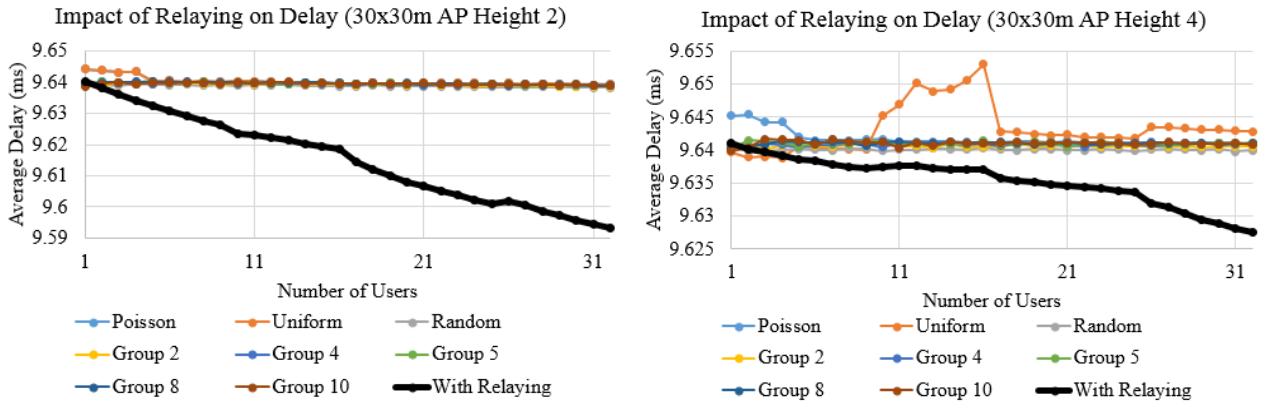


Figures 8 (left) and 9 (right). The average frame delay per user when there is no relaying in 10x10m VR playgrounds is shown with respect to the number of users for each distribution. Relaying performance is shown with maximal matching in bold. The delays are averaged over all distributions. Figure 8 on the left and figure 9 on the right show the results when the AP is placed at 2m and 4m height respectively.





Figures 10 (left) and 11 (right). The average frame delay per user in 20x20m VR playgrounds is shown with respect to the number of users for each distribution.



Figures 12 (left) and 13 (right). The average frame delay per user in 30x30m VR playgrounds is shown with respect to the number of users for each distribution.

The delays with and without relaying are shown. When there is no relaying, the average delay is independent of node distribution and number of users. This is because only delays of connected users are averaged. The exception is uniform distribution where average delay is contingent on grid layout and positions that are filled as the number of users increases.

Relaying decreases the average delay as the number of users increase because relaying also amplifies signals. Channels involved in relaying have higher capacities than direct LOS links due to signal attenuation. When there are more blocked users, relaying delivers frames more quickly than direct transmission in VR playgrounds that have fewer users. This indicates that relaying can improve QoE even when there are no blockages.

3.6 Discussion

The average data rate declines at consistently larger rates in figures 2, 4, and 6 than in figures 3, 5, and 7 respectively. This indicates that AP height plays a significant role in determining the amount of blockage and relaying performance in VR playgrounds. The inverse linear relationship between data rate and number of users persists in all cases. As the number of users increases, relaying yields increasing gains in connectivity. When the AP is placed at 2m height, relaying increases average data rate by 2.4%, 8.6%, and 10.2% in 10x10m, 20x20m, and 30x30m playgrounds with 4, 16, and 32 users respectively. This decreases to 0.2%, 1.3%, and 1.3% when AP height is raised to 4m.

There is little difference in average delays in all experiments. This is because in static relaying experiments, the delays of blocked nodes cannot be quantified. The impact of relaying on delay will become apparent in dynamic relaying experiments shown in subsequent sections. Nodes that are blocked or switch relays incur substantial delays.

4 Prolonging Network Stability in Dynamic Relay Virtual Reality Networks

Stable matching algorithms are necessary to stabilize relay VR networks. The impact of user mobility in relay VR networks is introduced in section 4.1 Section 4.2 clarifies the problem formulation for dynamic bipartite stable matching. The objective function to maximize network stability. Section 4.3 proposes stable matching algorithms. Section 4.4 is on experimental design. Section 4.5 illustrates the numerical results and section 4.6 discusses stability gains from stable matching algorithms.

4.1 Introduction

When node mobility is simulated with relay VR networks, link outages are unavoidable and frequent rerouting is needed to maintain user connectivity. Significant delays are incurred from relay switching. In all dynamic experiments but one, latency requirements cannot be met for at least one node. Stable routing decisions over time are necessary for users' QoE in VR playgrounds.

In this section, matching algorithms are modified to maximize network stability over time. The stable algorithms prioritize previously established pairs that are broken only when there are no free parents. The most frequently rerouted nodes are prioritized in matching. Multi-user VR applications are typically goal-oriented and team-based. Group random waypoint distributes users in groups and establishes structures in their movements that can be exploited in matching. Stable group matching algorithm explicitly uses this structure in matching to produce the most stable routing decisions in which latency requirements are met in all but on experiment. This makes stable group matching most suitable for stabilizing relay VR networks.

4.2 Dynamic Bipartite Matching Problem Formulation

Bipartite matching in relay VR networks is formulated at timestep t with inputs,

$$\boldsymbol{W} = \{1, .., n\}, \boldsymbol{V_t} \subset \boldsymbol{W}, f: LOS(x, y) \tag{7}$$

outputs,

$$\boldsymbol{S_t} = \{(1, p_{1,t}), (2, p_{2,t}), \dots, (n, p_{n,t})\}, where\ p_{i,t} \in \{1, \dots, n\}/\{i\} \tag{8}$$

and constraints,

$$for\ \forall p_{i,t} \neq \emptyset,\ p_{m,t} \neq i\ for\ \forall m \in \boldsymbol{W}, \tag{9}$$

$$p_{i,t} \neq p_{j,t} \; for \; \forall i,j \in \boldsymbol{W}, i \neq j \tag{10}$$

Under user mobility, $\boldsymbol{V}$ set membership and LOS changes. Network stability is important for maintaining VR application quality. The objective function is to maximize network stability over time by maximizing the number of preserved output pairs in $\boldsymbol{S}$ across t timesteps,

$$\text{maximize} \sum_{\boldsymbol{t}} |\boldsymbol{S_t} \cap \boldsymbol{S_{t-1}}| \tag{11}$$

4.3 Stable Matching Algorithms

Greedy matching, maximal matching, stable matching, and group matching are modified to prioritize established links. The algorithms in section 3.3 are used on the first timestep as there are no previously established links. In subsequent timesteps, previous pairs are maintained if LOS requirements are met and the most instable nodes are prioritized in matching. When the algorithms fail to match blocked nodes with free parents, algorithm 5 is used to find the blocked node's most preferred parent among established pairs whose child has LOS to the AP and has been least frequently rerouted. Algorithm 5 relies on the rerouting frequency of each node i, denoted by δ_i. The terms 'pair', 'link', and 'match' are used interchangeably.

ALGORITHM 5: Function: Node **Find_parent**(Node v, Set **U**)

Input: Blocked Node v, Sorted Candidate Parent Node Set **U**

Output: Free Parent Node p

1 : Parent $p = \emptyset$;

2 : For each node u in **U**: //Find parent with least δ

3 : If $p_{u,t} \neq \emptyset$:

4 : If $p = \emptyset$ or $\delta_u < \delta_p$:

5 : $p = u$;

6 : Else if $\exists i \in \boldsymbol{W} - \boldsymbol{V_t}$ s.t. $p_{i,t} = u$:

7 : If $p = \emptyset$ or $\delta_i < \delta_p$:

8 : $p = i$;

9 : If $p \neq \emptyset$: //Break the pair containing desired parent

10: If $p_{u,t} \neq \emptyset$:

11: $p_{u,t} = \emptyset$;

12: Else:

13: $p_{i,t} = \emptyset$;

14: Return p; //Return the parent for matching

The algorithm takes in the sorted set of candidate parents of node v, and returns the most preferred and least frequently rerouted parent that can be used for matching.

4.3.1 Stable Greedy Matching

ALGORITHM 6: Stable Greedy Matching

1 : For each node *w* in **W**: //Match previous pairs if LOS satisfied
2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:
3 : $p_{w,t} = p_{w,t-1}$;
4 : Sort $\boldsymbol{V_t}$ by δ in descending order; //Sort nodes by rerouting frequency
5 : For each node *v* in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$:
6 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t}$ s.t. for $\forall u \in \boldsymbol{U_v}$: $LOS(v,u)$ and $LOS(u, AP)$; //Candidate parents
7 : Sort $\boldsymbol{U_v}$ by *v*'s preference; //Sort by link capacities
8 : For each node *u* in $\boldsymbol{U_v}$: //Match most preferred free parent with highest capacity
9 : If $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$:
10: $p_{v,t} = u$;
11: Break;
12: If $p_{v,t} = \emptyset$:
13: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_v}$);

Stable greedy matching prioritizes established pairs in each timestep by first re-pairing them in lines 1-3. The list of blocked nodes is sorted by descending δ in line 4 and traversed for matching in lines 5-11. Blocked nodes are made to prefer parents with higher capacities and are greedily matched with their most preferred parent. If no free parents remain, algorithm 5 is used to find a free parent from remaining candidate parents in lines 12-13. Breaking previously established pairs improves connectivity at the cost of relaying stability.

4.3.2 Stable Maximal Matching

ALGORITHM 7: Stable Maximal Matching

1 : For each node *w* in **W**: //Match previous pairs if LOS satisfied
2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:
3 : $p_{w,t} = p_{w,t-1}$;
4 : Sort $\boldsymbol{V_t}$ by δ in descending order; //Sort nodes by rerouting frequency
5 : For each node *v* in $\boldsymbol{V_t}$:
6 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t}$ s.t. for $\forall u \in \boldsymbol{U_v}$: $LOS(v,u)$ and $LOS(u, AP)$; //Candidate parents

7 : Sort $\boldsymbol{U_v}$ by v's preference; //Sort by link capacities

8 : For each node v in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$:

9 : For each node u in $\boldsymbol{U_v}$: //Match most preferred free parent with highest capacity

10: If $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$:

11: $p_{v,t} = u$;

12: Else if $\exists i \in \boldsymbol{V_t}$ s.t. $p_{i,t} = u$, $p_{i,t} \neq p_{i,t-1}$, and i can be assigned a new parent j:

13: $p_{i,t} = j, p_{v,t} = u$;

14: If $p_{v,t} = \emptyset$:

15: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_v}$);

In lines 1-3, established pairs are re-paired granted LOS requirements are satisfied. Lines 4-7 sort the list of blocked nodes in order of descending δ and each nodes' candidate parent list according to their preference. Maximal matching is performed with free parents in lines 8-13, and lines 14-15 finds a free parent from the list of candidate parents if no free parents remain.

4.3. Stable Gale-Shapley Matching

Stable Gale-Shapley matching does not necessarily produce stable pairings. The algorithm prioritizes established links over stable pairings in order to prolong network stability. Although sub-optimally stable pairings are formed, these pairings are practically identical to stable pairings in highly dynamic VR networks.

ALGORITHM 8: Stable Gale-Shapley Matching

---

1 : For each node w in **W**: //Match previous pairs if LOS satisfied

2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:

3 : $p_{w,t} = p_{w,t-1}$;

4 : For each node v in $\boldsymbol{V_t}$:

5 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t}$ s.t. for $\forall u \in \boldsymbol{U_v}$: $LOS(v,u)$ and $LOS(u, AP)$; //Candidate Parents

6 : Sort $\boldsymbol{U_v}$ according to v's preference; //Sort by link capacity

7 : While $\exists v$ in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$ and $\exists q \in \boldsymbol{W} - \boldsymbol{V_t}$ s.t. $p_{i,t} \neq q$ for $\forall i \in \boldsymbol{W}$:

8 : If $\exists u \in \boldsymbol{U_v}$ s.t. $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$: //Match free parents

9 : $p_{v,t} = u$;

10: Else if $\exists i \in \boldsymbol{V_t}$ s.t. $p_{i,t} = u, p_{i,t} \neq p_{i,t-1}$:

11: If u prefers v over i: //Prefer node with higher link capacity

12: $p_{v,t} = u, p_{i,t} = \emptyset$;

13: Else:

14: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_v}$);

Previous pairs are preserved in lines 1-3. Lines 4-6 sort the list of candidate parents for each blocked node according to their preference. Stable matching among free parents is performed in lines 7-12, and algorithm 5 is used to find a free parent from remaining candidate parents if there are no free parents left in lines 13-14.

4.3.4 Stable Group Matching

Stable group matching first preserves previous pairs, then performs matching among group members, followed by matching among non-group members, and finally uses algorithm 5 when no free parents are available. The algorithm utilizes node groups. A node group i is defined as $\boldsymbol{G_i} \subseteq \boldsymbol{W}$ for $i \in \{1, \dots, m\}$, where m is the total number of groups, $\boldsymbol{G_i} \cap \boldsymbol{G_j} = \emptyset$ for $\forall i, j \in \{1, \dots, m\}, i \neq j$, and $\boldsymbol{G_1} \cup \boldsymbol{G_2} \cup \cdots \cup \boldsymbol{G_m} = \boldsymbol{W}$.

ALGORITHM 9: Stable Group Matching

---

1 : For each node w in **W**: //Match previous pairs if LOS satisfied

2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:

3 : $p_{w,t} = p_{w,t-1}$;

4 : Let $v \in \boldsymbol{G_{vj}}$ for $j \in \{1, \dots, m\}$; //$\boldsymbol{G_{vj}}$ denotes v's group

5 : For each node v in $\boldsymbol{V_t}$: //Sort in-group and out-group parent sets

6 : Let $\boldsymbol{U_{g,v}} \subseteq \boldsymbol{G_{vj}} - \boldsymbol{V_t}$ and $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t} - \boldsymbol{U_{g,v}}$ where for $\forall u \in \boldsymbol{U_{g,v}}$ and $\forall u \in \boldsymbol{U_v}$:

6 : $LOS(v,u)$ and $LOS(u, AP)$;

7 : Sort $\boldsymbol{U_{g,v}}$ and $\boldsymbol{U_v}$ according to v's preference; //Sort by link capacity

8 : While $\exists v$ in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$ and $\exists q \in \boldsymbol{W} - \boldsymbol{V_t}$ s.t. $p_{i,t} \neq q$ for $\forall i \in \boldsymbol{W}$:

9 : If $\exists u \in \boldsymbol{U_{g,v}}$ s.t. $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$://In-group matching

10: $p_{v,t} = u$;

11: Else if $\exists u \in \boldsymbol{U_{g,v}}$ s.t. $p_{u,t} = \emptyset$, and $\exists i$ s.t. $p_{i,t} = u$ and $p_{i,t} \neq p_{i,t-1}$:

12: If $i \in \boldsymbol{V_t} - \boldsymbol{G_{vj}}$ or u prefers v over i: //Prefer node with higher link capacity

13: $p_{i,t} = \emptyset, p_{v,t} = u$;

14: Else if $\exists u \in \boldsymbol{U_v}$ s.t. $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$://Out of group matching

15: $p_{v,t} = u$;

16: Else if $\exists u \in \boldsymbol{U_v}$ s.t. $p_{u,t} = \emptyset$, and $\exists i \in \boldsymbol{V_t} - \boldsymbol{G_{u,j}}$ s.t. $p_{i,t} = u$ and $p_{i,t} \neq p_{i,t-1}$:

17: If u prefers v over i: //Prefer node with higher link capacity

18: $p_{v,t} = u, p_{i,t} = \emptyset$;

19: Else:

20: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_{g,v}} \cup \boldsymbol{U_v}$);

Previous pairs are preserved in lines 1-3, and two candidate parent lists are formed and sorted in lines 5-7. $U_{g,v}$ contains candidate parents in the same group as the blocked node; whereas, U_v contains the remaining candidate parents. Stable matching among group members and non-group members is performed in lines 8-13 and 14-18 respectively. A candidate parent is assigned in lines 19-20 when there are no free parents left.

4.4 Dynamic Experiment Design

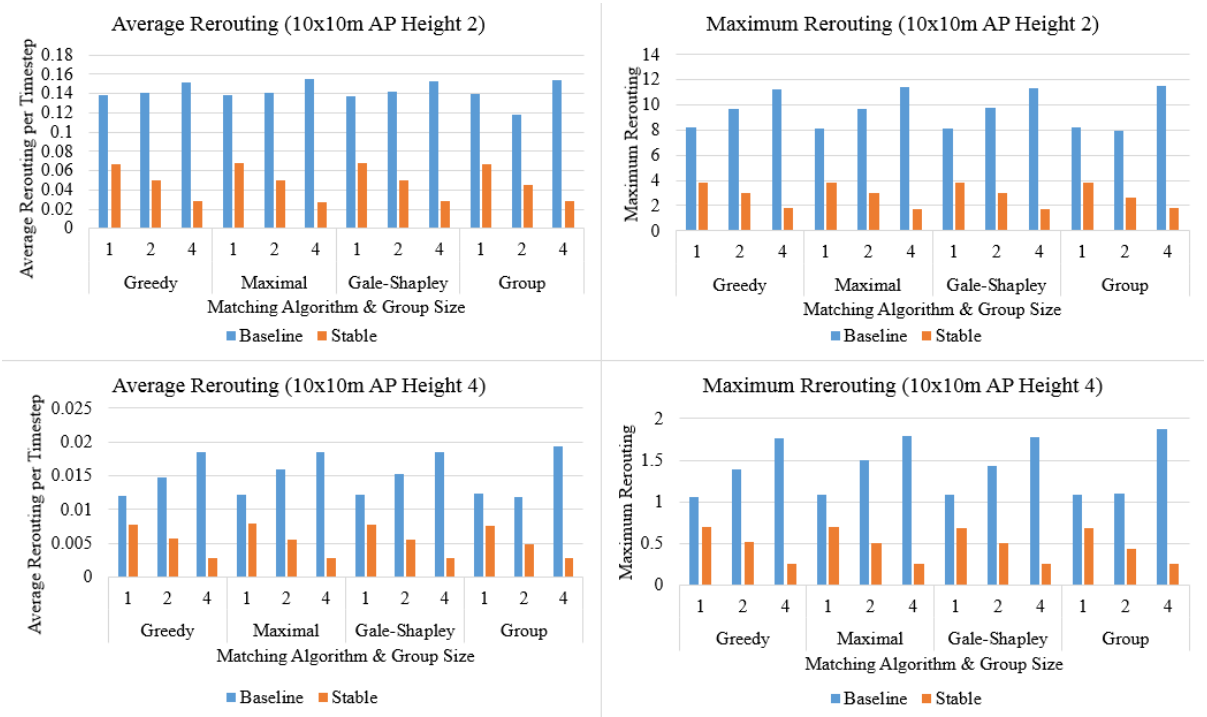
Group random waypoint [4] is used to model user mobility. In experiments, user movements are relatively more chaotic than in reality. This is suitable for evaluating network stability as links are often broken. User movements in multi-user VR games are usually goal-orientated and contain some structure that are captured in group distributions.

Nodes are initially distributed using group random distribution described in section 3.4. 4, 16, and 32 users are simulated in 10x10m, 20x20m, and 30x30m playgrounds respectively with the AP at 2m or 4m height. The group's movement is determined by the group leader's movement which follows the random waypoint model. In random waypoint, a node's movement cycles between moving towards a randomly chosen destination on the shortest path with constant velocity, and remaining stationary. Group leaders to move at most 1m horizontally and vertically in each timestep and randomly distribute their idle duration between 0 and 5 timesteps. Group members maintain their initial relative positions to the group leader. 10,000 trials of 100 timesteps each are performed in each experiment. The average is taken over all group sizes.

4.5 Numerical Results

The numerical results are divided into three sections for each playground. Each section quantifies the impact of matching algorithms on rerouting frequency and QoE.

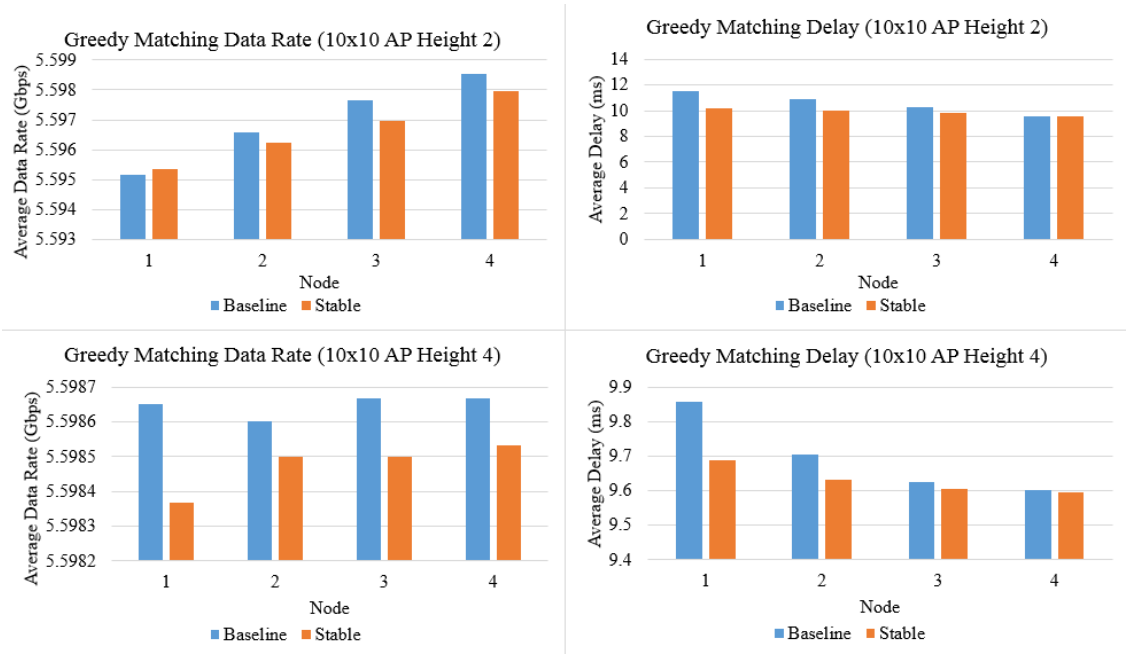
4.5.1 10x10m Virtual Reality Playgrounds Results



Figures 14 (top left), 15 (top right), 16 (bottom left), and 17 (bottom right). The average rerouting frequency per timestep is shown on the left, and the maximum rerouting frequency is shown on the right. AP height is 2m for the top figures and 4m for the bottom figures.

All matching algorithms proposed in section 4.3 significantly reduce rerouting incurred by their baseline counterparts. The proposed algorithms benefit from increasing group size because communication channels tend to be more reliable when nodes maintain their relative position to one another. Baseline matching algorithms generally prioritizes the highest capacity link. These links change depending on the group's formation and orientation relative to the AP. In large groups and small playgrounds, the highest capacity link changes frequently and accounts for the increase in rerouting frequency.

AP height contributes immensely to rerouting frequency. A difference of 2m in AP height leads to roughly a 10-fold difference in rerouting frequency. The proposed algorithms reduce rerouting frequency by 65.6% and 61.4% on average from baseline algorithms when the AP is placed at 2m and 4m height respectively. The maximum rerouting frequency is reduced by 68.8% and 61.8% for corresponding AP heights.

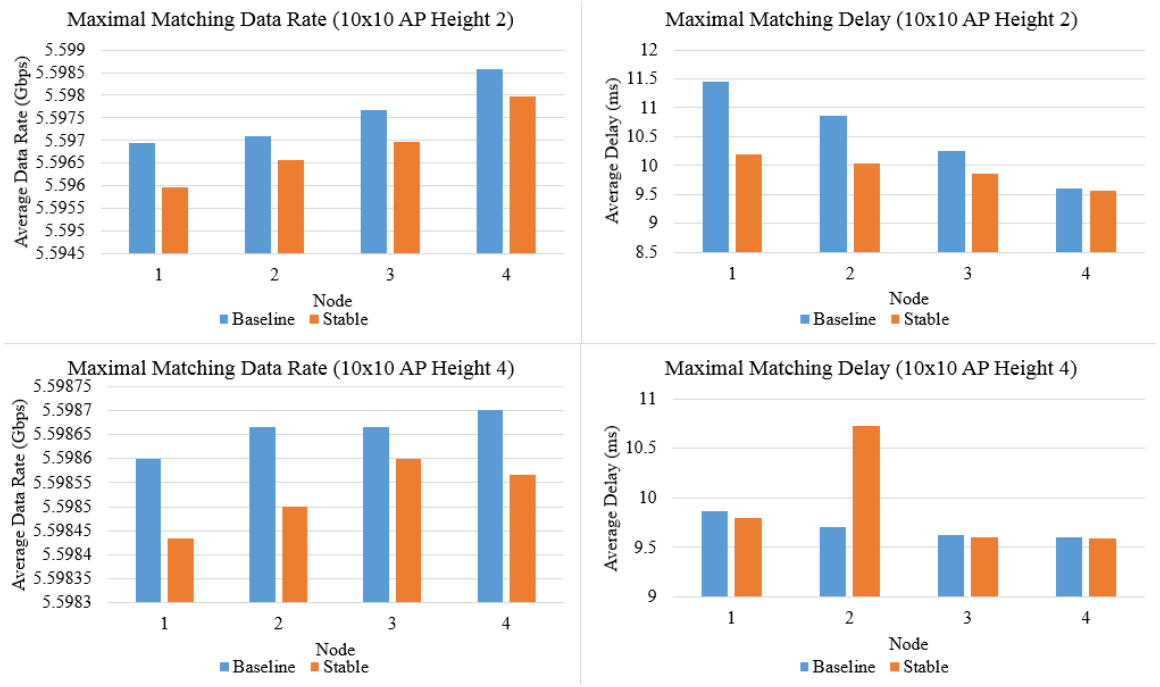


Figures 18 (top left), 19 (top right), 20 (bottom left), and 21 (bottom right). The QoE metrics of each node for greedy matching are graphed with average data rate on the left and average delay on the right. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies each node with shorter nodes taking smaller identifiers and taller taking larger identifiers.

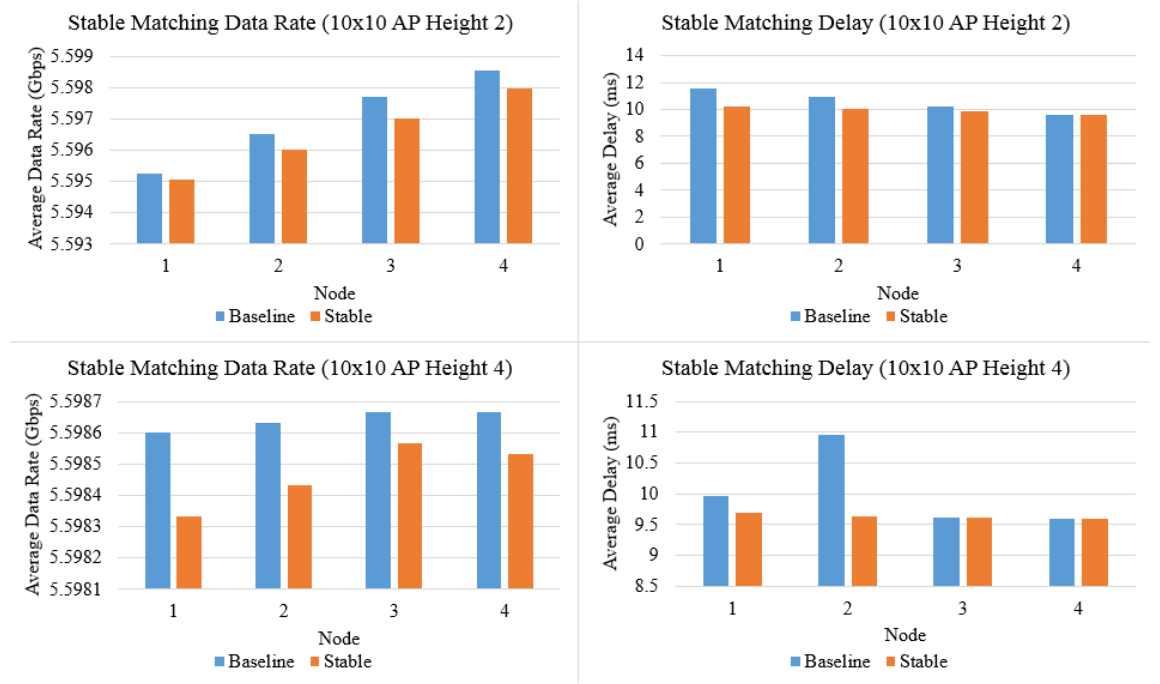
The average data rate generally decreases when established links are prioritized because preserving previous pairings is suboptimal to forming the most preferred pairing in terms of matching. Stable matching algorithms fail to match nodes slightly more often than baseline algorithms. On the other hand, signal attenuation is negligible in small 10x10m playgrounds because all channel capacities support the required data rate. The proposed algorithms improve average delay despite there being additional delays from more matching failures. This indicates that reducing rerouting frequency improves QoE in relay VR networks.

When the AP is elevated by 2m, average data rates and delays are improved. Similarly, taller nodes have superior data rates and delays. At increased heights, there is less blockage and greater FOV. Blockage and rerouting are directly related; whereas, FOV is directly related to the number of candidate relays. Matching algorithms tend to fail to match shorter nodes with fewer candidate relays. Fair matching algorithms are developed in section 5 to tackle this matching disparity.

The discussion for greedy matching also applies to maximal, stable, and group matching. Discrepancies are due to randomness in experiments

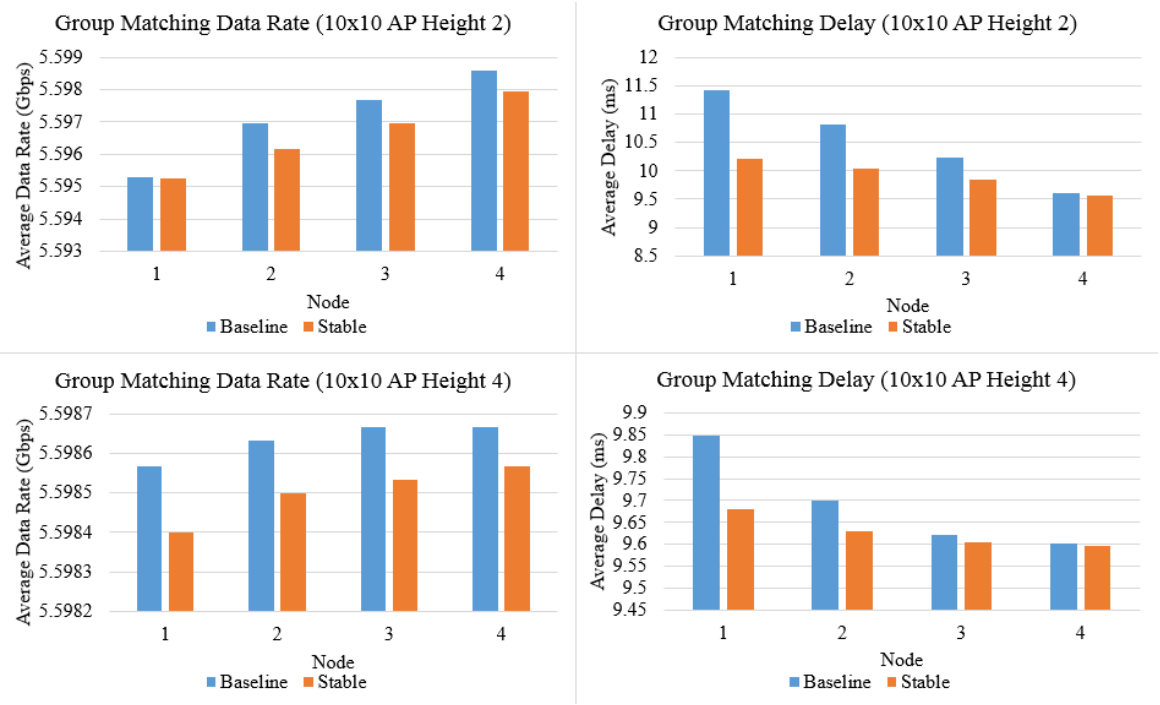


Figures 22 (top left), 23 (top right), 24 (bottom left), and 25 (bottom right). The average data rate and delay for maximal matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.



Figures 26 (top left), 27 (top right), 28 (bottom left), and 29 (bottom right). The average data rate and delay for stable matching is shown on the left and right respectively. The AP is located

at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.

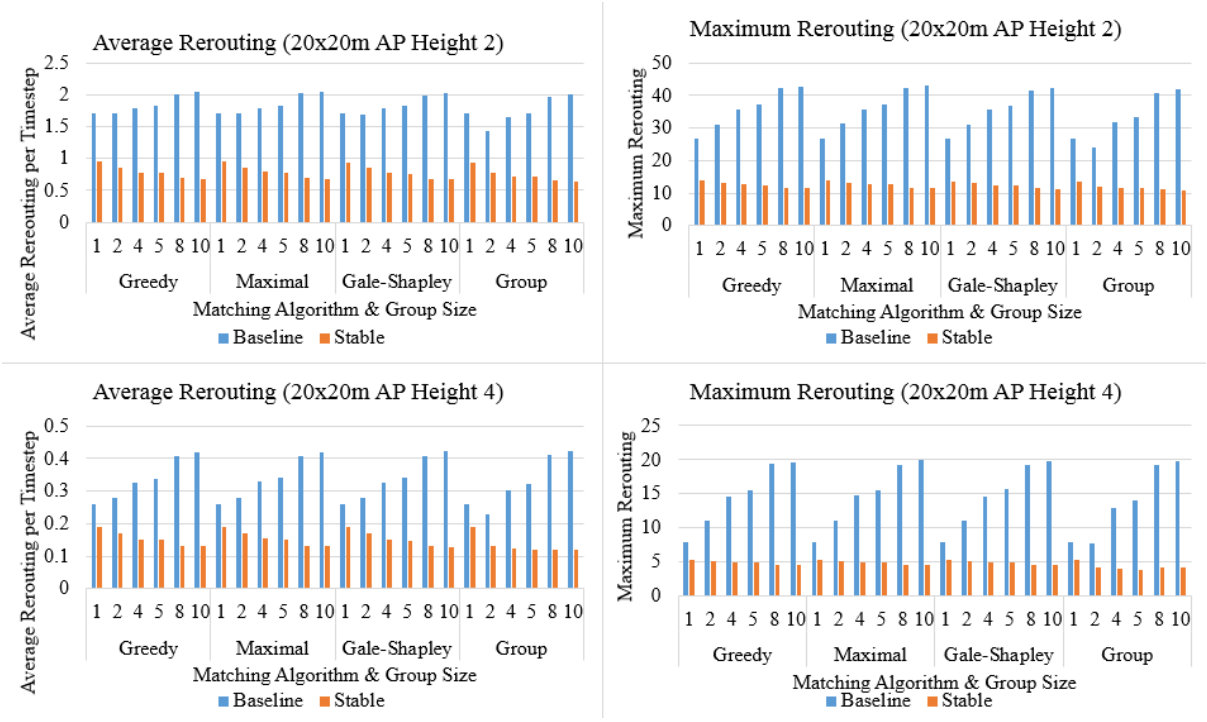


Figures 30 (top left), 31 (top right), 32 (bottom left), and 33 (bottom right). The average data rate and delay for group matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.

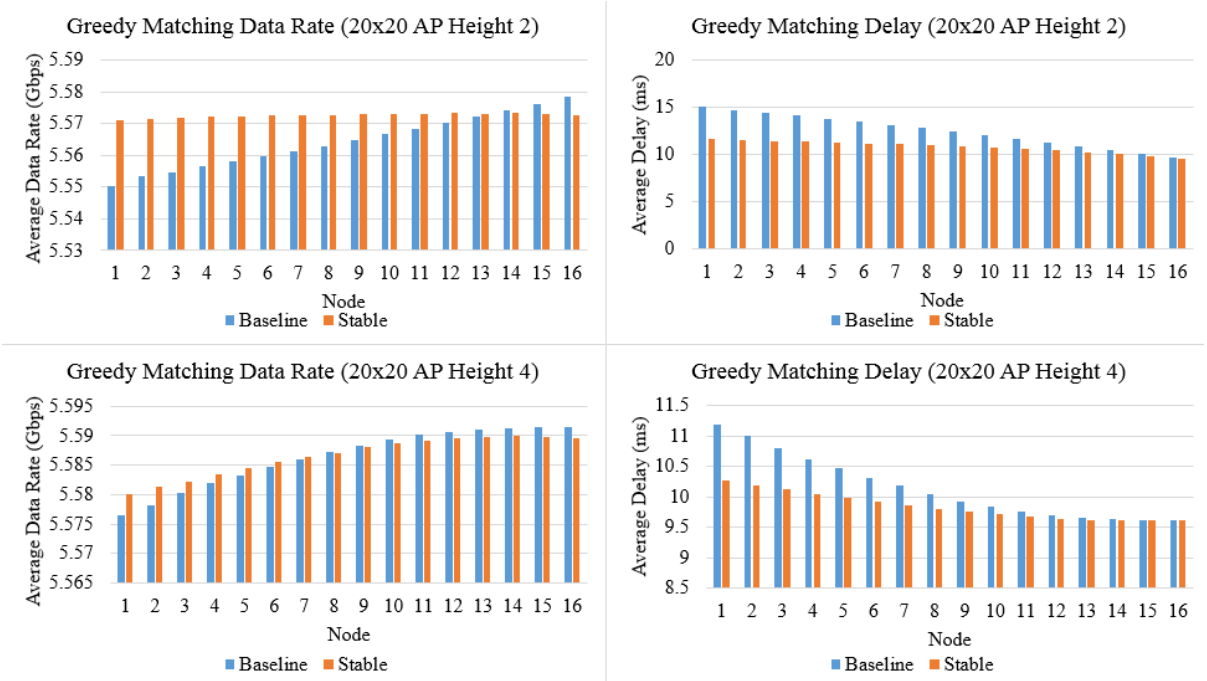
4.5.2 20x20m Virtual Reality Playgrounds Results

The results in 20x20m playgrounds mirror the results from 10x10m playgrounds except average data rates for shorter nodes are consistently higher for stable matching algorithms. Stable algorithms reduce average rerouting frequency by 56.7% and 60.0% on average for 2m and 4m AP height respectively. Maximum rerouting frequency is reduced by 63.4% and 62.5%.

When 20x20m playgrounds are simulated with 16 users, blockage is more prevalent and there are considerably more combinations of node configurations than in 10x10m playgrounds with 4 users. The frequency of rerouting scales with the playground, and shorter nodes experience increasingly frequent rerouting. When nodes with the highest rerouting frequencies are prioritized matching, the same nodes that tend to fail to be matched are prioritized. Since shorter nodes also experience poorer connectivity, stable matching algorithms redistribute connectivity from taller to shorter nodes and provide fairer QoE to all users. This applies to both AP heights, but the effect is less pronounced when the AP is placed at 4m height.

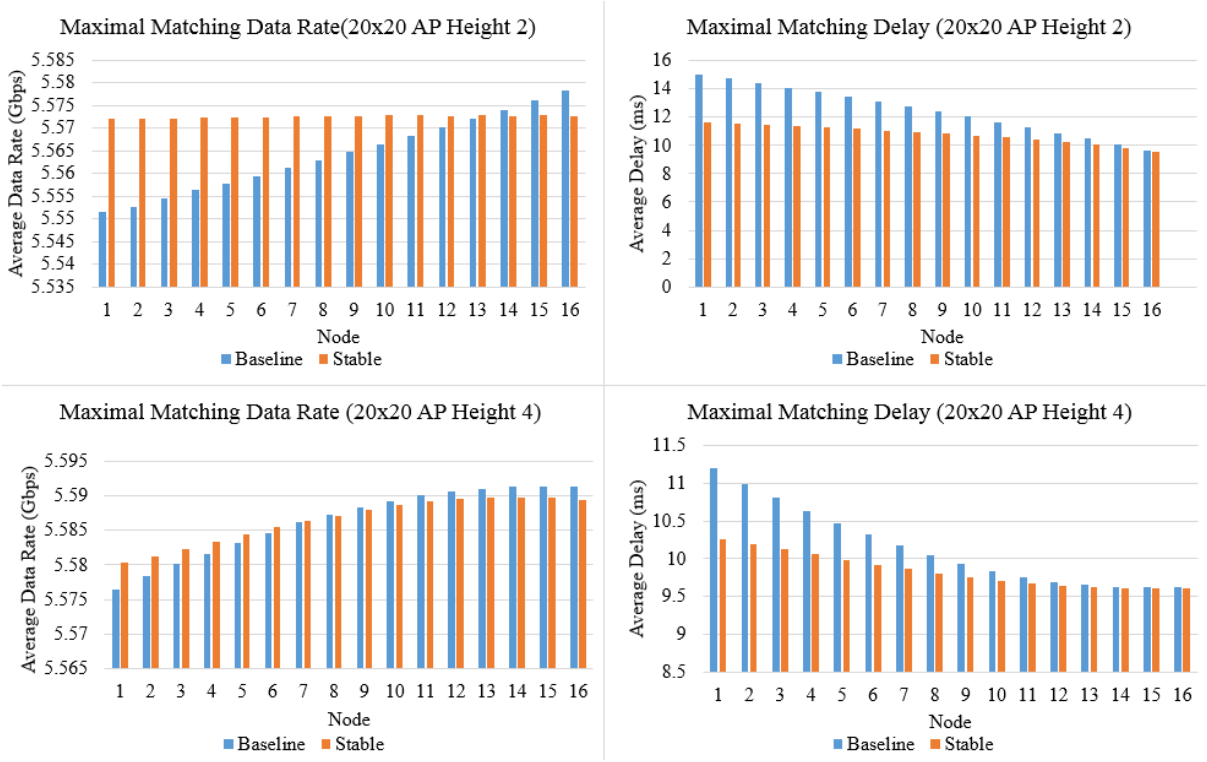


Figures 34 (top left), 35 (top right), 36 (bottom left), and 37 (bottom right). The average rerouting frequency per timestep is shown on the left, and the maximum rerouting frequency is shown on the right. AP height is 2m for the top figures and 4m for the bottom figures.



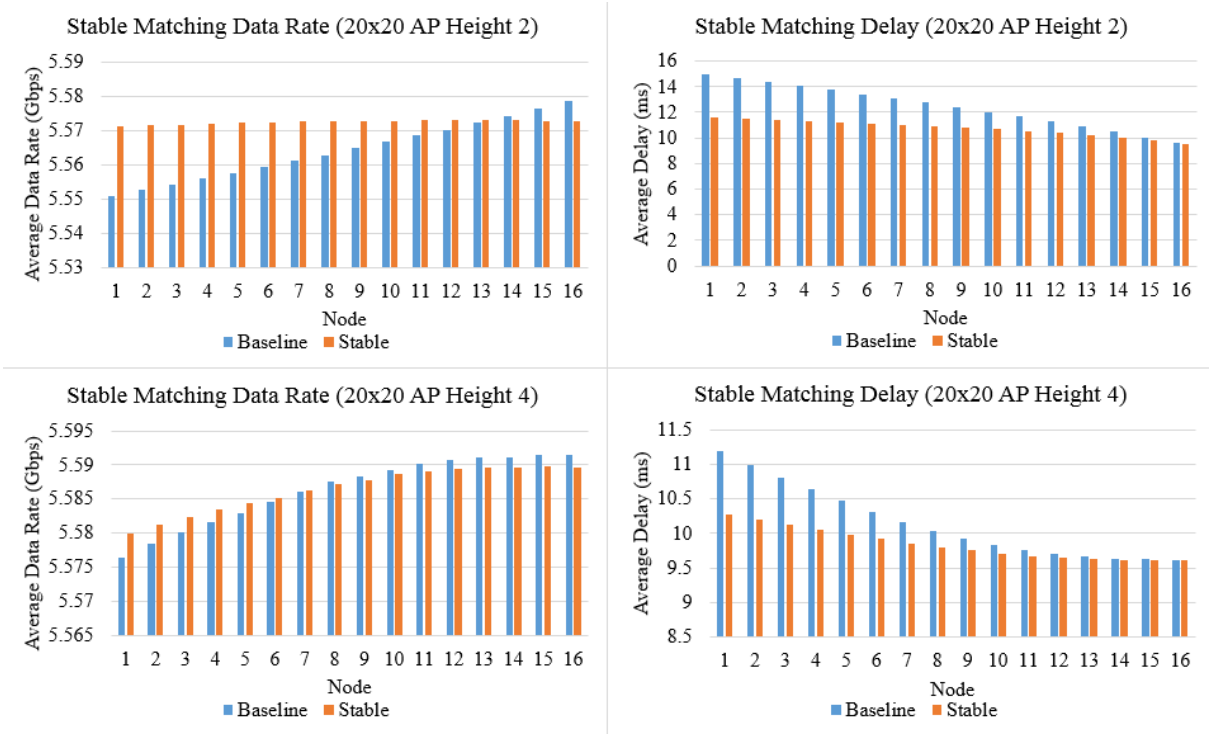
Figures 38 (top left), 39 (top right), 40 (bottom left), and 41 (bottom right). The average data rate and delay for greedy matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.

Figure 38 shows that stable greedy matching improves overall average data rate when the AP is located at 2m height. When the most frequently rerouted node, and thereby the shortest node, is prioritized in greedy matching, the same node which has the fewest candidate relays is prioritized. This allows for more nodes to be matched. Overall average data rate improves because the neediest nodes are greedily matched with their most preferred relay first.



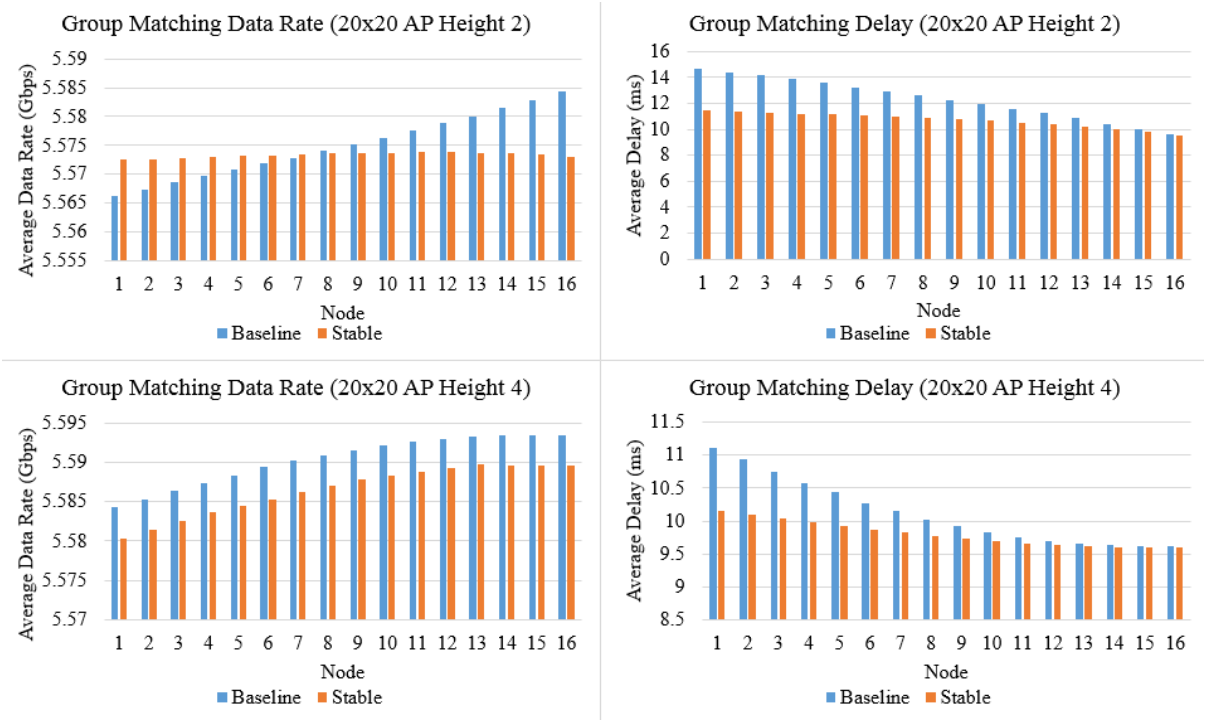
Figures 42 (top left), 43 (top right), 44 (bottom left), and 45 (bottom right). The average data rate and delay for maximal matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.

Figure 42 shows that stable maximal matching increases overall average data rate despite encountering more matching failures. Stable maximal matching does not always produce the maximal match because it prioritizes previous pairings. In contrast, maximal matching greedily matches a blocked node with its most preferred relay in a way that maximizes the number of pairings. The capacities of links formed between pairings are not considered. When the most frequently rerouted nodes are paired with the highest capacity candidate relay, all other pairings can be maximally matched in a way that improves overall average data rate. QoE is maximized by maximizing link capacities on top of the number of pairings. The impact of signal attenuation cannot be ignored in large VR playgrounds.



Figures 46 (top left), 47 (top right), 48 (bottom left), and 49 (bottom right). The average data rate and delay for stable matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.

Figure 46 shows that stable Gale-Shapley matching improves overall average data rate. This is due to a loophole in the algorithm that allows frequently rerouted nodes to obtain high capacity relays. Specifically, stable Gale-Shapley matching does not maintain pairing stability because prior pairings are preserved. Algorithm 5 allows blocked nodes to bypass relay nodes' preference and break prior pairs to obtain their most preferred relay. In subsequent timesteps, these pairings are protected because they have been established. This loophole allows frequently rerouted nodes who are less preferred by relays to communicate on high capacity links. The pairs formed by the loophole allows more matches to be made, leading to fewer matching failures and improving overall data rate.

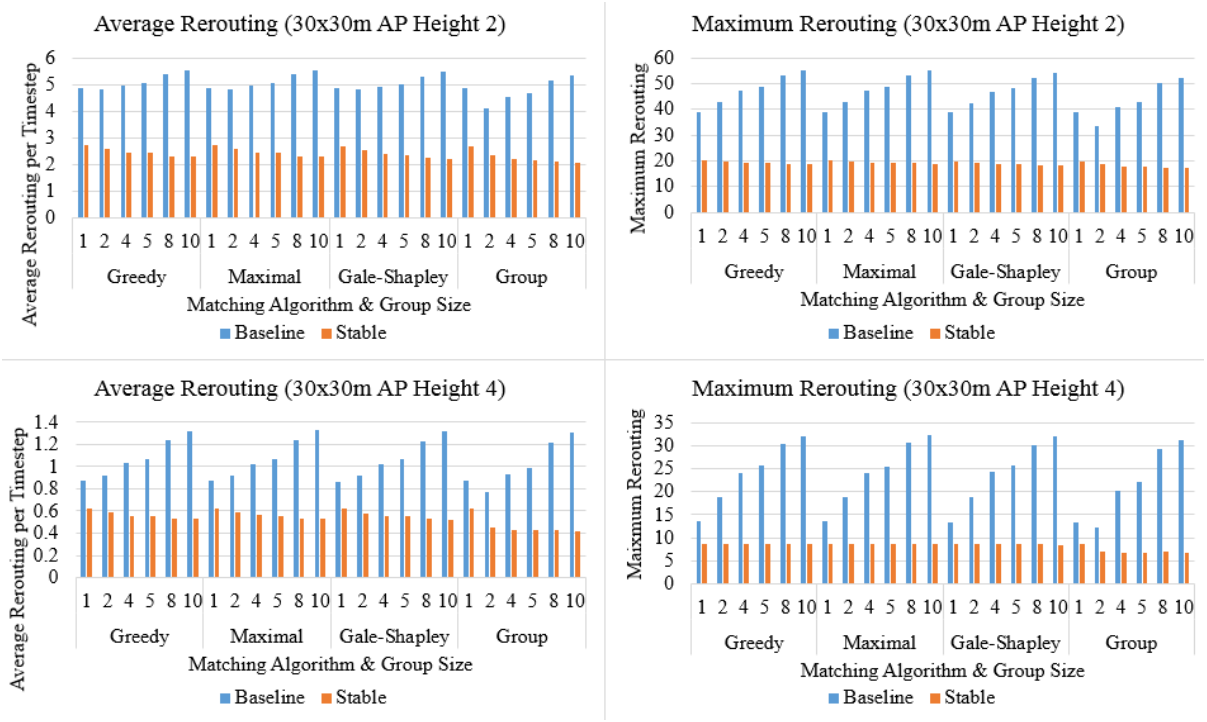


Figures 50 (top left), 51 (top right), 52 (bottom left), and 53 (bottom right). The average data rate and delay for group matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures. The x-axis identifies nodes with shorter nodes taking smaller identifiers.

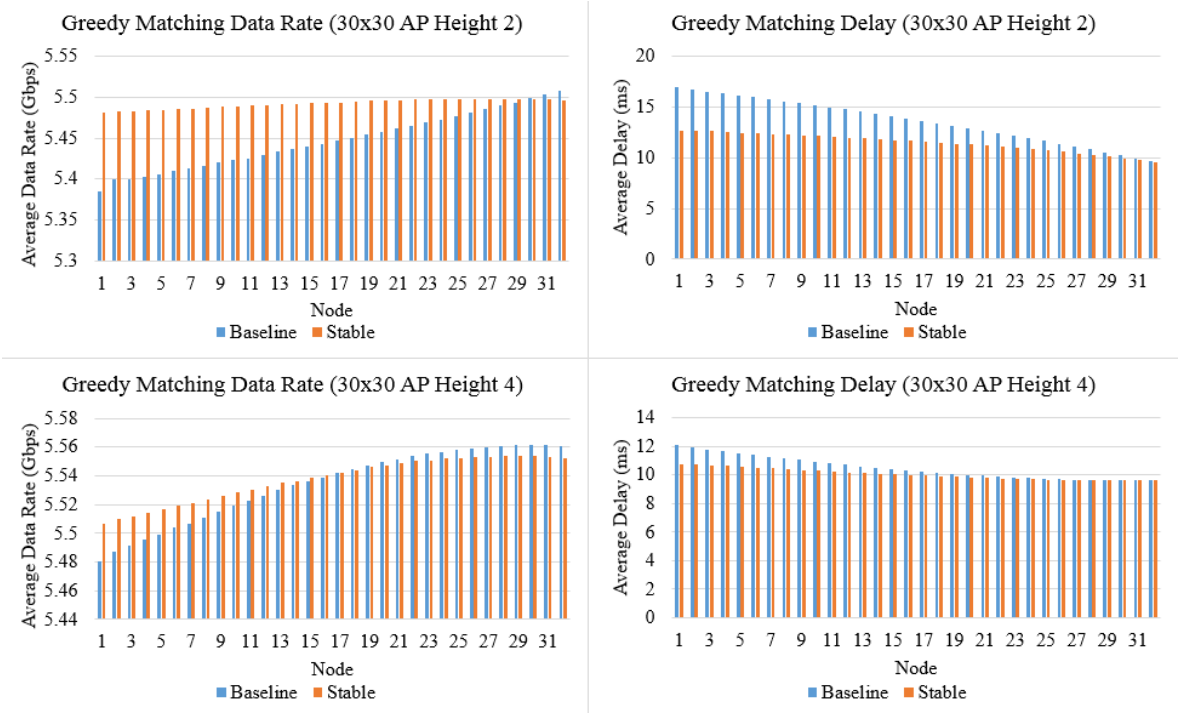
Stable group matching redistributes connectivity and leads to slightly lower overall average data rate when the AP is at 2m height. Figure 52 shows that stable group matching consistently leads to lower data rates when the AP is at 4m height. The results are expected because established links are suboptimal. Data rate is sacrificed to avoid delays incurred by relay switching.

4.5.3 30x30m Virtual Reality Playgrounds Results

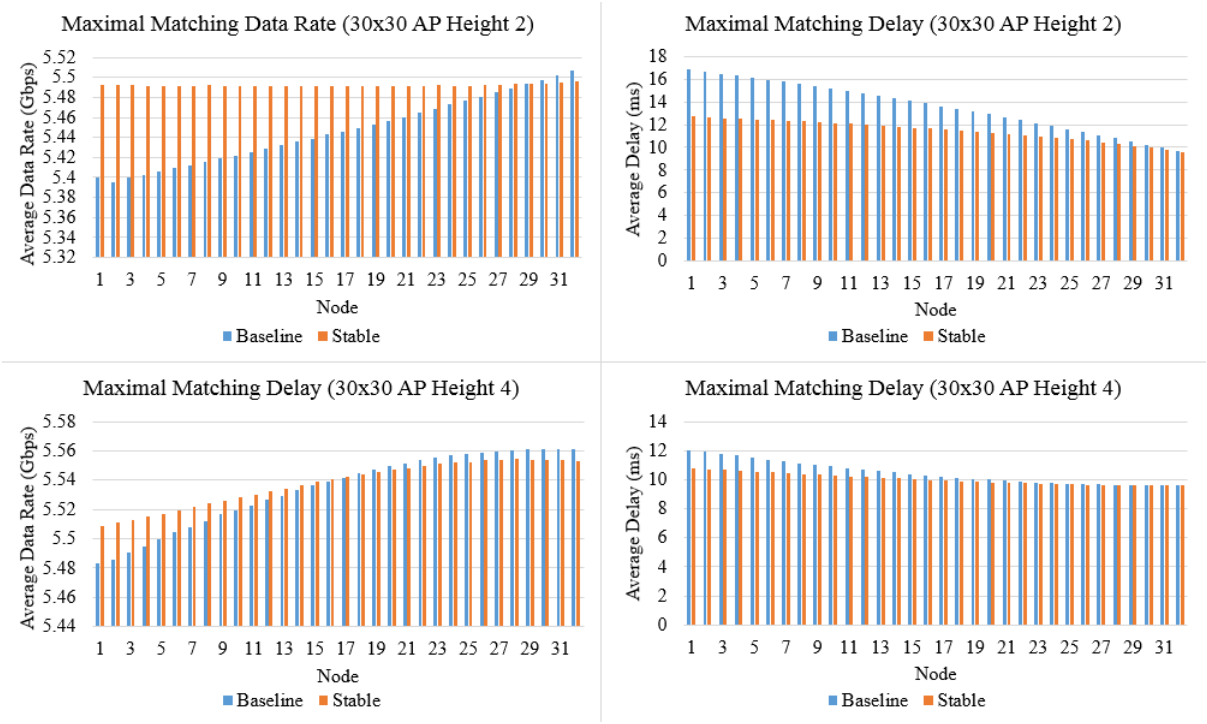
The results in 30x30m playgrounds are identical to the result from 20x20m playgrounds relative to 10x10m playgrounds. Stable matching algorithms reduce rerouting by 52.0% and 47.5% on average when the AP is placed at 2m and 4m height respectively. Maximum rerouting frequency is reduced by 58.4% and 61.2%.



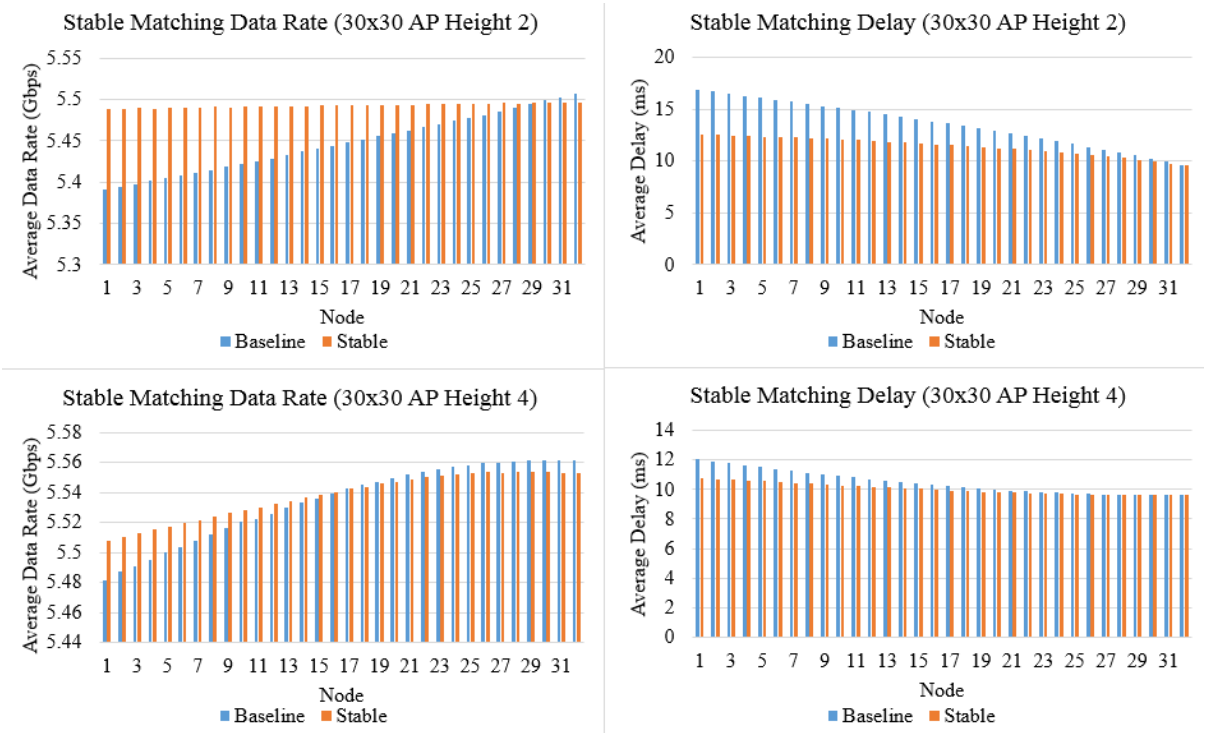
Figures 54 (top left), 55 (top right), 56 (bottom left), and 57 (bottom right). The average rerouting frequency per timestep is shown on the left, and the maximum rerouting frequency is shown on the right. AP height is 2m for the top figures and 4m for the bottom figures.



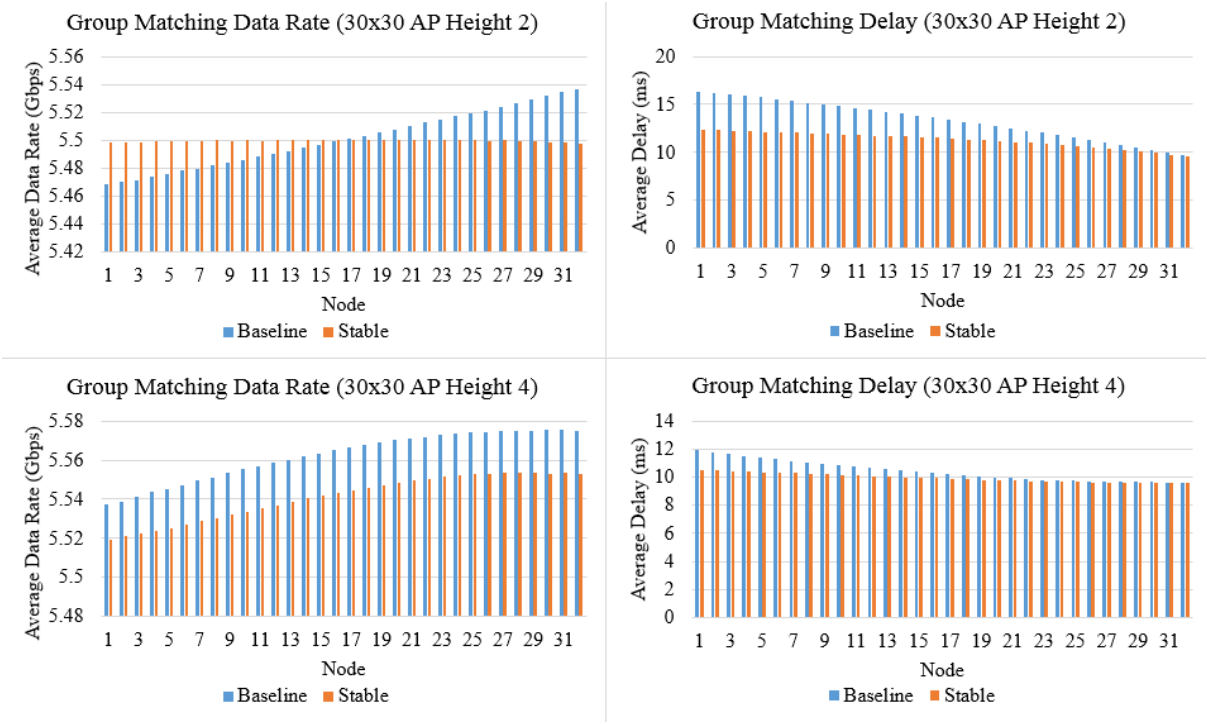
Figures 58 (top left), 59 (top right), 60 (bottom left), and 61 (bottom right). The average data rate and delay for greedy matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures.



Figures 62 (top left), 63 (top right), 64 (bottom left), and 65 (bottom right). The average data rate and delay for maximal matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures.



Figures 66 (top left), 67 (top right), 68 (bottom left), and 69 (bottom right). The average data rate and delay for stable matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures.



Figures 70 (top left), 71 (top right), 72 (bottom left), and 73 (bottom right). The average data rate and delay for group matching is shown on the left and right respectively. The AP is located at 2m height for the top figures, and 4m height for the bottom figures.

4.6 Discussion

Matching Algorithm	Average Rerouting per Timestep	
	AP Height 2	AP Height 4
Stable Greedy	0.048697	0.005428
Stable Maximal	0.048333	0.005391
Stable Gale-Shapley	0.048426	0.005326
Stable Group	0.047114	0.005062

Table 3. Average rerouting in 10x10m playgrounds.

Matching Algorithm	Average Rerouting per Timestep	
	AP Height 2	AP Height 4
Stable Greedy	0.789764	0.153456
Stable Maximal	0.790871	0.153629
Stable Gale-Shapley	0.781495	0.153089
Stable Group	0.740590	0.133727

Table 4. Average rerouting in 20x20m playgrounds.

Matching Algorithm	Average Rerouting per Timestep	
	AP Height 2	AP Height 4
Stable Greedy	2.454284	0.563676
Stable Maximal	2.458070	0.564125
Stable Gale-Shapley	2.394201	0.559171
Stable Group	2.255968	0.462493

Table 5. Average rerouting in 30x30m playgrounds.

Matching Algorithm	Average Data Rate		Average Delay	
	AP Height 2	AP Height 4	AP Height 2	AP Height 4
Stable Greedy	5.5966	5.5985	9.9223	9.6311
Stable Maximal	5.5969	5.5985	9.9087	9.9280
Stable Gale-Shapley	5.5965	5.5985	9.9259	9.6312
Stable Group	5.5966	5.5985	9.9130	9.6278

Table 6. Average data rate and delay in 10x10m playgrounds.

Matching Algorithm	Average Data Rate		Average Delay	
	AP Height 2	AP Height 4	AP Height 2	AP Height 4
Stable Greedy	5.5725	5.5865	10.7810	9.8389
Stable Maximal	5.5725	5.5865	10.7817	9.8393
Stable Gale-Shapley	5.5724	5.5865	10.7681	9.8383
Stable Group	5.5733	5.5865	10.7001	9.8022

Table 7. Average data rate and delay in 20x20m playgrounds.

Matching Algorithm	Average Data Rate		Average Delay	
	AP Height 2	AP Height 4	AP Height 2	AP Height 4
Stable Greedy	5.4921	5.5375	11.4604	10.7808
Stable Maximal	5.4922	5.5376	11.4633	10.7846
Stable Gale-Shapley	5.4925	5.5375	11.4110	10.0433
Stable Group	5.4999	5.5411	11.2957	9.95476

Table 8. Average data rate and delay in 30x30m playgrounds.

The performance of all stable matching algorithms in every experiment is summarized in the tables above. Stable matching algorithms are necessary for stabilizing relay VR networks. Baseline algorithms have difficulty meeting latency requirements even in the smallest of playgrounds. The proposed algorithms only fail to meet latency requirements in 30x30m playgrounds when the AP is placed at 2m height. If the AP is placed at 4m height, 32 users can be supported within latency requirements. The average data rate decreases as the scale of VR playgrounds increase chiefly due to matching failures. Prediction algorithms can be developed to anticipate and handle matching failures. Stable group matching outperforms all other algorithms and can be used to support the largest number of users in relay VR networks.

5 Guaranteeing Fair Connectivity in Dynamic Relay Virtual Reality Networks

Fair matching algorithms are devised to equally distribute QoS to all users. The disparity of QoE in relay VR networks is introduced in section 5.1. Section 5.2 formulates the objective function for maximizing connectivity fairness. Stable matching algorithms are modified in section 5.3. The experimental parameters are provided in section 5.4. Experimental results are explained in section 5.5 and discussed in section 5.6.

5.1 Introduction

Relaying in VR networks fails to provide connectivity to all nodes in some scenarios when the AP is placed at 2m height. In particular, matching algorithms repeatedly fail to match the same particular nodes over time. These nodes are repeatedly disconnected, and QoS is unfair. Fair connectivity is achieved by equally distributing QoS. In fair relay VR networks, several nodes may be briefly disconnected instead of disconnecting only one or two nodes regularly. The algorithms in section 4.3 are tweaked to achieve improvements in connectivity fairness by prioritizing nodes that have experienced the greatest amount of disconnect in matching. Simulations show that the proposed algorithms achieve fair relaying in VR networks.

5.2 Dynamic and Fair Bipartite Matching Problem Formulation

Bipartite matching in relay VR networks is formulated at timestep t with inputs,

$$\boldsymbol{W} = \{1, .., n\}, \boldsymbol{V_t} \subset \boldsymbol{W}, f: LOS(x, y) \tag{7}$$

outputs,

$$\boldsymbol{S_t} = \{(1, p_{1,t}), (2, p_{2,t}), \dots, (n, p_{n,t})\}, where\ p_{i,t} \in \{1, \dots, n\}/\{i\} \tag{8}$$

and constraints,

$$for\ \forall p_{i,t} \neq \emptyset,\ p_{m,t} \neq i\ for\ \forall m \in \boldsymbol{W}, \tag{9}$$

$$p_{i,t} \neq p_{j,t}\ for\ \forall i, j \in \boldsymbol{W}, i \neq j \tag{10}$$

In section 4, the objective function is,

$$\text{maximize} \sum_{\boldsymbol{t}} |\boldsymbol{S_t} \cap \boldsymbol{S_{t-1}}| \tag{11}$$

When perfect matches cannot be found, existing algorithms often fail to match the same particular nodes. As a result, QoS becomes unevenly distributed. The objective functions of this section is to maximize connectivity fairness by minimizing the maximum number of times

a node is disconnected in t timesteps. Suppose that the number of timesteps a node i is disconnected is,

$$\gamma_i = \sum_t (LOS_t(AP, i) = \emptyset) \cap (p_{i,t} = \emptyset)$$

The objective function for maximizing connectivity fairness is,

$$\text{minimize} \max_{i \in W} \gamma_i \quad (12)$$

5.3 Fair Matching Algorithms

Stable matching algorithms are modified to maximize connectivity fairness. The nodes with the highest γ_i in each timestep are prioritized in matching. Greedy and maximal matching matches nodes with the highest γ_i first by sorting the list of blocked nodes in lines 4 of the respective algorithms. Stable and group matching algorithms are identical to the ones proposed in the previous section, except parent nodes prefer nodes with higher γ_i.

5.3.1 Fair Greedy Matching

ALGORITHM 10: Fair Greedy Matching

1 : For each node w in $\mathbf{W}$: //Match previous pairs if LOS satisfied
2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:
3 : $p_{w,t} = p_{w,t-1}$;
4 : Sort $\boldsymbol{V_t}$ by γ in descending order;
5 : For each node v in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$:
6 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t}$ s.t. for $\forall u \in \boldsymbol{U_v}: LOS(v, u)$ and $LOS(u, AP)$; //Candidate parents
7 : Sort $\boldsymbol{U_v}$ by v's preference; //Sort by link capacity
8 : For each node u in $\boldsymbol{U_v}$: //Match most preferred free parent with highest capacity
9 : If $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$:
10: $p_{v,t} = u$;
11: Break;
12: If $p_{v,t} = \emptyset$:
13: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_v}$);

Fair greedy matching prioritizes established pairs in each timestep by first re-pairing them in lines 1-3. The list of blocked nodes is sorted by descending γ in line 4 and traversed for matching in lines 5-11. Blocked nodes are made to prefer closer parents and are greedily

matched with their most preferred parent. If no free parents remain, algorithm 5 is used to find a free parent from remaining candidate parents in lines 12-13.

5.3.2 Fair Maximal Matching

Although maximal matching achieves maximum connectivity, there are often multiple maximal matchings in relay VR networks. In Ford-Fulkerson algorithm, blocked nodes are traversed in the same order. The algorithm produces maximal matchings that fail to match the same particular nodes. This matching bias is remedied by traversing the list of blocked nodes in order of descending γ_i. When maximal matching cannot produce a perfect match, nodes that have historically experienced the most trouble are matched first.

ALGORITHM 11: Fair Maximal Matching

---

1 : For each node w in $\mathbf{W}$: //Match previous pairs if LOS satisfied

2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:

3 : $p_{w,t} = p_{w,t-1}$;

4 : Sort $\boldsymbol{V_t}$ by γ in descending order;

5 : For each node v in $\boldsymbol{V_t}$:

6 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t}$ s.t. for $\forall u \in \boldsymbol{U_v}$: $LOS(v,u)$ and $LOS(u, AP)$; //Candidate parents

7 : Sort $\boldsymbol{U_v}$ by v's preference; //Sort by link capacity

8 : For each node v in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$:

9 : For each node u in $\boldsymbol{U_v}$: //Match most preferred free parent

10: If $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$:

11: $p_{v,t} = u$;

12: Else if $\exists i \in \boldsymbol{V_t}$ s.t. $p_{i,t} = u$, $p_{i,t} \neq p_{i,t-1}$, and i can be assigned a new parent by 9-13:

13: $p_{v,t} = u$;

14: If $p_{v,t} = \emptyset$:

15: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_v}$);

In lines 1-3, established pairs are repaired granted LOS requirements are satisfied. Lines 4-7 sort the list of blocked nodes in order of descending γ and each nodes' candidate parent list according to their preference. Maximal matching is performed with free parents in lines 8-13, and lines 14-15 finds a free parent from the list of candidate parents if no free parents remain.

5.3.3 Fair Stable Matching

Stable Gale-Shapley matching is made fair by incorporating γ_i in parent's preference. Specifically, parents are made to prefer blocked nodes with higher γ_i. If γ_i is the same for two blocked nodes, then the closer node is preferred.

ALGORITHM 12: Fair Stable Matching

1 : For each node *w* in **W**: //Match previous pairs if LOS satisfied
2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:
3 : $p_{w,t} = p_{w,t-1}$;
4 : For each node *v* in $\boldsymbol{V_t}$:
5 : Let $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t}$ s.t. for $\forall u \in \boldsymbol{U_v}$:$LOS(v,u)$ and $LOS(u, AP)$; //Candidate parents
6 : Sort $\boldsymbol{U_v}$ according to *v*'s preference; //Sort by link capacity
7 : While $\exists v$ in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$ and $\exists q \in \boldsymbol{W} - \boldsymbol{V_t}$ s.t. $p_{i,t} \neq q$ for $\forall i \in \boldsymbol{W}$:
8 : If $\exists u \in \boldsymbol{U_v}$ s.t. $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$: //Match most preferred free parent
9 : $p_{v,t} = u$;
10: Else if $\exists i \in \boldsymbol{V_t}$ s.t. $p_{i,t} = u, p_{i,t} \neq p_{i,t-1}$:
11: If u prefers *v* over *i*: //Prefer node with larger γ
12: $p_{v,t} = u, p_{i,t} = \emptyset$;
13: Else:
14: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_v}$);

Previous pairs are preserved in lines 1-3. Lines 4-6 sort the list of candidate parents for each blocked node according to their preference. Stable matching among free parents is performed in lines 7-12, and algorithm 5 is used to find a free parent from remaining candidate parents if there are no free parents left in lines 13-14.

5.3.4 Fair Group Matching

Group matching is made fair by having parents prefer nodes with higher γ after group membership is considered. Distance is used for further tie-breaking.

ALGORITHM 13: Fair Group Matching

1 : For each node *w* in **W**: //Match previous pairs if LOS satisfied
2 : If $LOS(p_{w,t-1}, AP)$ and $LOS(w, p_{w,t-1})$:
3 : $p_{w,t} = p_{w,t-1}$;

4 : Let $v \in \boldsymbol{G_{vj}}$ for $j \in \{1, \ldots, m\}$; //$\boldsymbol{G_{vj}}$ denotes v's group

5 : For each node v in $\boldsymbol{V_t}$: //Sort in-group and out-group parent sets

6 : Let $\boldsymbol{U_{g,v}} \subseteq \boldsymbol{G_{vj}} - \boldsymbol{V_t}$ and $\boldsymbol{U_v} \subseteq \boldsymbol{W} - \boldsymbol{V_t} - \boldsymbol{U_{g,v}}$ where for $\forall u \in \boldsymbol{U_{g,v}}$ and $\forall u \in \boldsymbol{U_v}$:

6 : $LOS(v,u)$ and $LOS(u,AP)$;

7 : Sort $\boldsymbol{U_{g,v}}$ and $\boldsymbol{U_v}$ according to v's preference; //Sort by link capacity

8 : While $\exists v$ in $\boldsymbol{V_t}$ s.t. $p_{v,t} = \emptyset$ and $\exists q \in \boldsymbol{W} - \boldsymbol{V_t}$ s.t. $p_{i,t} \neq q$

8 : for $\forall i \in \boldsymbol{W}$: //Stable matching in-group first then out-group

9 : If $\exists u \in \boldsymbol{U_{g,v}}$ s.t. $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$:

10: $p_{v,t} = u$;

11: Else if $\exists u \in \boldsymbol{U_{g,v}}$ s.t. $p_{u,t} = \emptyset$, and $\exists i$ s.t. $p_{i,t} = u$ and $p_{i,t} \neq p_{i,t-1}$:

12: If $i \in \boldsymbol{V_t} - \boldsymbol{G_{vj}}$ or u prefers v over i: //Prefer node with larger γ

13: $p_{i,t} = \emptyset, p_{v,t} = u$;

14: Else if $\exists u \in \boldsymbol{U_v}$ s.t. $p_{u,t} = \emptyset$ and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$:

15: $p_{v,t} = u$;

16: Else if $\exists u \in \boldsymbol{U_v}$ s.t. $p_{u,t} = \emptyset$, and

16: $\exists i \in \boldsymbol{V_t} - \boldsymbol{G_{u,j}}$ s.t. $p_{i,t} = u$ and $p_{i,t} \neq p_{i,t-1}$:

17: If u prefers v over i: //Prefer node with larger γ

18: $p_{v,t} = u, p_{i,t} = \emptyset$;

19: Else:

20: $p_{v,t} =$ **Find_parent**(v, $\boldsymbol{U_{g,v}} \cup \boldsymbol{U_v}$);

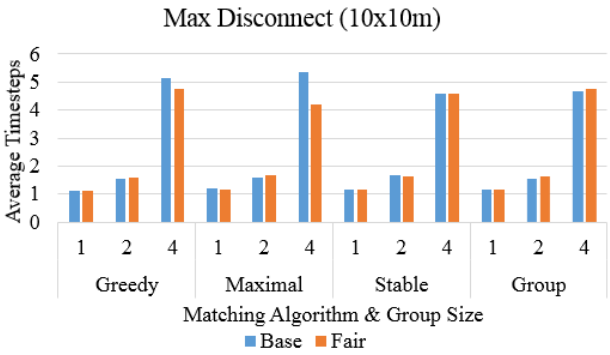
Previous pairs are preserved in lines 1-3, and two candidate parent lists are formed and sorted in lines 5-7. $\boldsymbol{U_{g,v}}$ contains candidate parents in the same group as the blocked node; whereas, $\boldsymbol{U_v}$ contains the remaining candidate parents. Stable matching among group members and non-group members is performed in lines 8-13 and 14-18 respectively. A candidate parent is assigned in lines 19-20 when there are no free parents left.

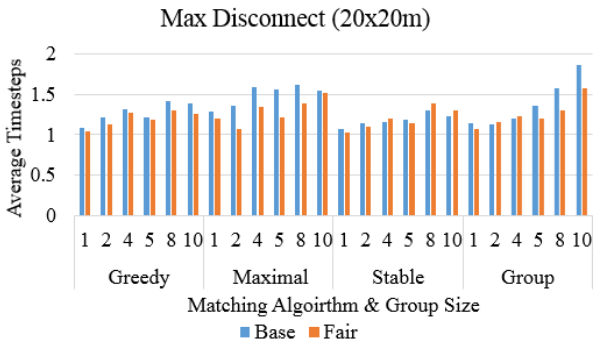
5.4 Dynamic Experiment Design

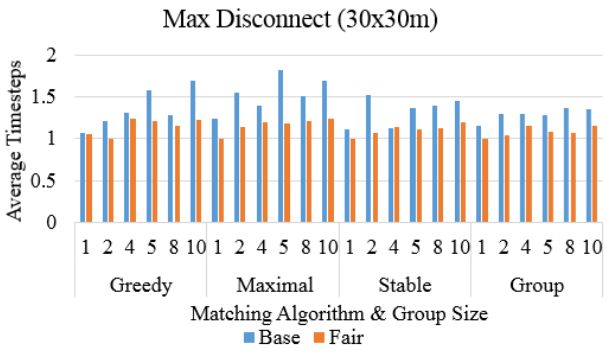
Group random waypoint [4] is used to model user mobility. Nodes are distributed in groups as described in section 3.4. The group's movement is determined by the group leader's movement which follows the random waypoint model. 4, 16, and 32 users are simulated in 10x10m, 20x20m, and 30x30m playgrounds. The AP is placed at 2m height in experiments. Matching failures do not occur when the AP is placed at 4m height. Group leaders move at

most 1m horizontally and vertically in each timestep and remain stationary for at most 5 timesteps. Group members maintain their initial relative positions to the group leader. 10,000 trials involving 100 timesteps each are performed for in each experiment. The average is taken over all group sizes.

5.5 Numerical Results

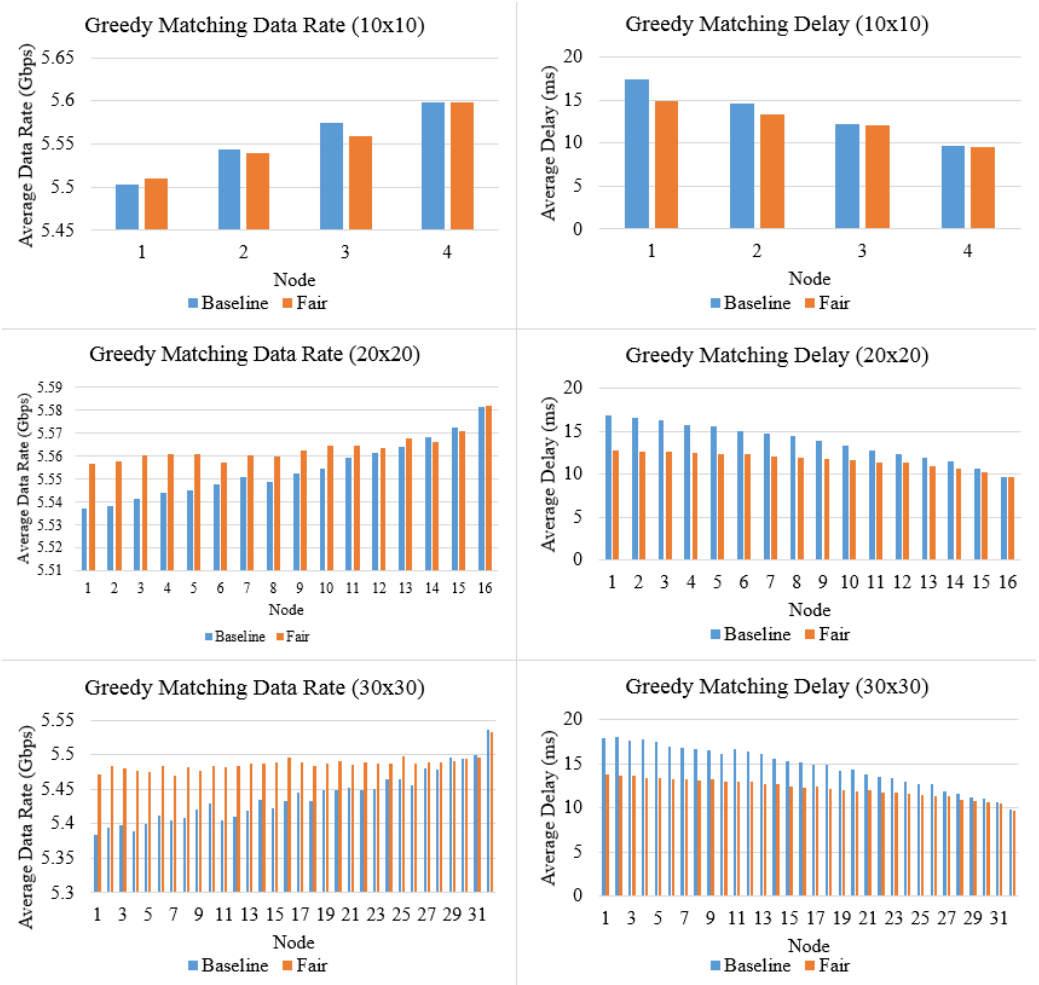






Figures 74 (top left), 75 (top right), and 76 (bottom left). The average maximum number of timesteps a node is disconnected in trials with matching failures is graphed for each algorithm and respective group size.

The max disconnect graphs above show that fair matching algorithms reduce the maximum number of timesteps a node is disconnected as playgrounds scale in size and number of users. When there are more users, there are more relays to reroute signals. All blocked nodes compete for relays and prioritizing nodes that have been more frequently disconnected ensures that the most frequently disconnected node will not be disconnected anymore unless it is no longer the most frequently disconnected node. The algorithm greedily maximizes connectivity fairness.

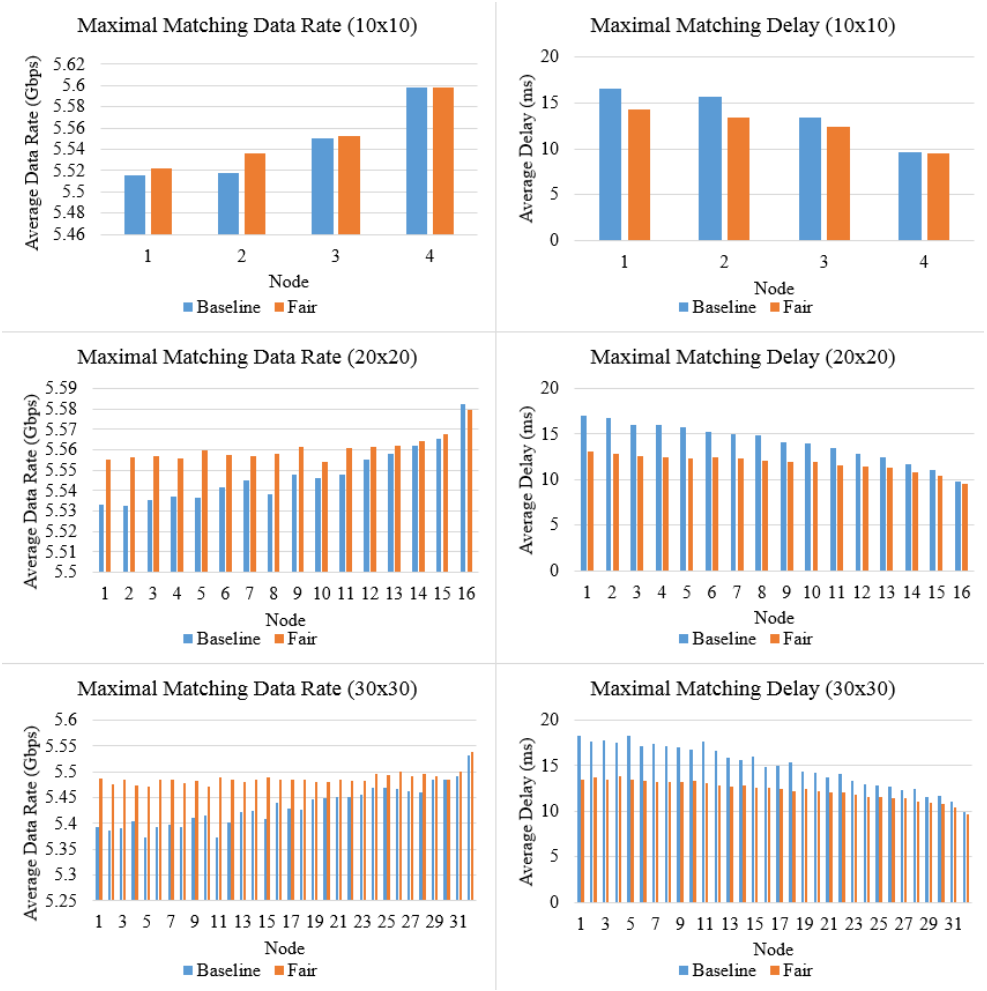


Figures 77 (top left), 78 (top right), 79 (middle left), 80 (middle right), 81 (bottom left), and 82 (bottom right). The average data rate and delay are shown for each experiment in which greedy matching fails to produce perfect matches in at least one timestep.

Matching Algorithm	Standard Deviation		
	10x10	20x20	30x30
Baseline	0.03581	0.01232	0.03607
Stable	0.03324	0.00487	0.00948
Fair	0.03173	0.00611	0.01047
Percent Reduction	11.3828	50.3853	91.6601

Table 9. Standard deviation of data rates achieved by baseline, stable, and fair greedy matching algorithms. The percent reduction is calculated with (baseline – fair) / baseline.

Greedy matching allows the proposed greedy method to work directly in matching. Nodes that have been disconnected the most are guaranteed to be matched if any relays are in LOS. This continues at all costs until other nodes are as equally disconnected. Fair greedy matching sacrifices more connectivity than is gained.

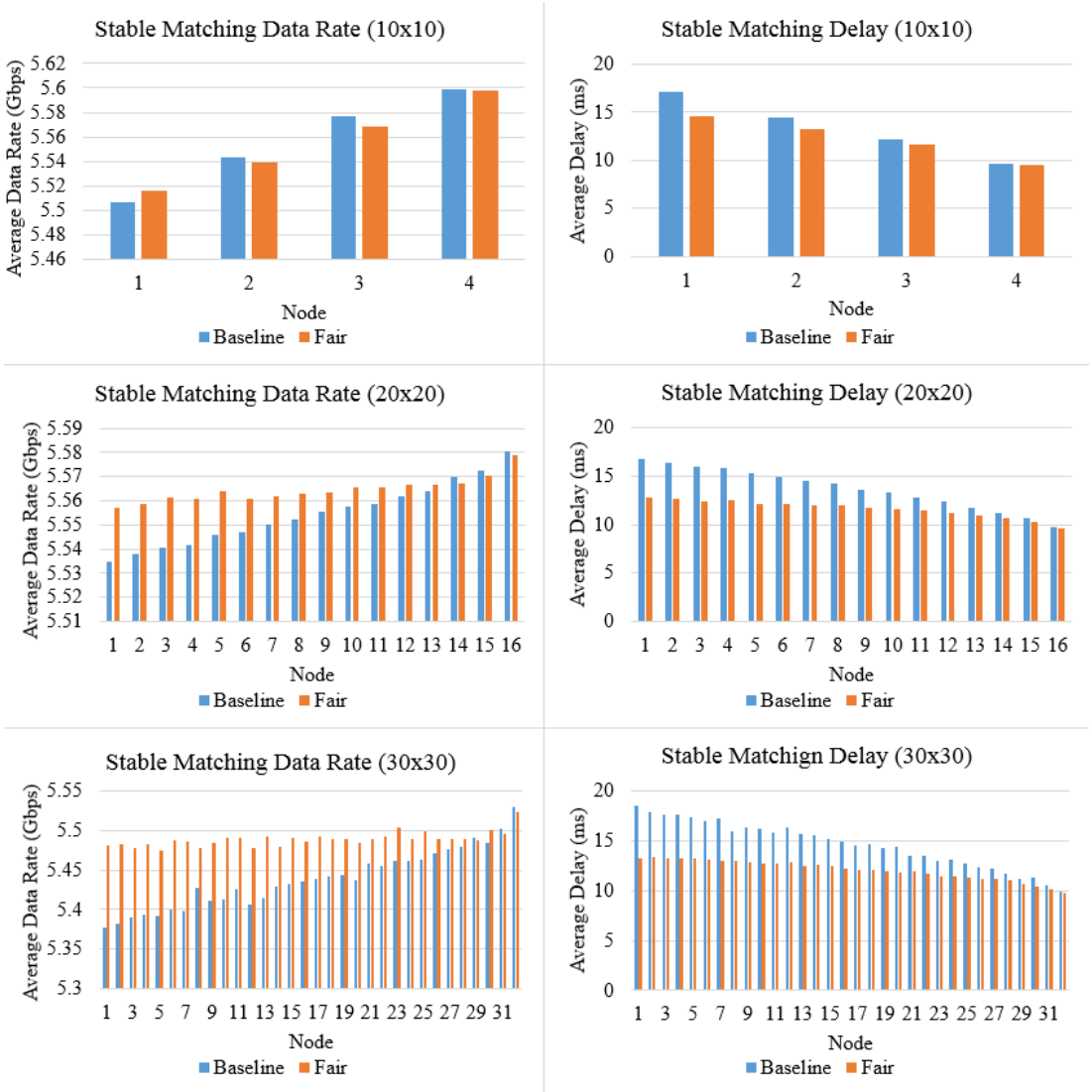


Figures 83 (top left), 84 (top right), 85 (middle left), 86 (middle right), 87 (bottom left), and 88 (bottom right). The average data rate and delay are shown for each experiment in which maximal matching fails to produce perfect matches in at least one timestep.

Matching Algorithm	Standard Deviation		
	10x10	20x20	30x30
Baseline	0.03363	0.01340	0.03813
Stable	0.03517	0.00557	0.01067
Fair	0.02842	0.00603	0.01152
Percent Reduction	15.4766	55.0161	69.7832

Table 10. Standard deviation of data rates achieved by baseline, stable, and fair maximal matching algorithms. The percent reduction is calculated with (baseline – fair) / baseline.

Fair maximal matching ensures that the most frequently disconnected nodes are matched when maximizing the number of pairings. The standard deviation in table 10 is larger for users in larger playgrounds because frequently disconnected nodes are often not paired with their most preferred relay. This pairing configuration allows for greater gains in overall connectivity.

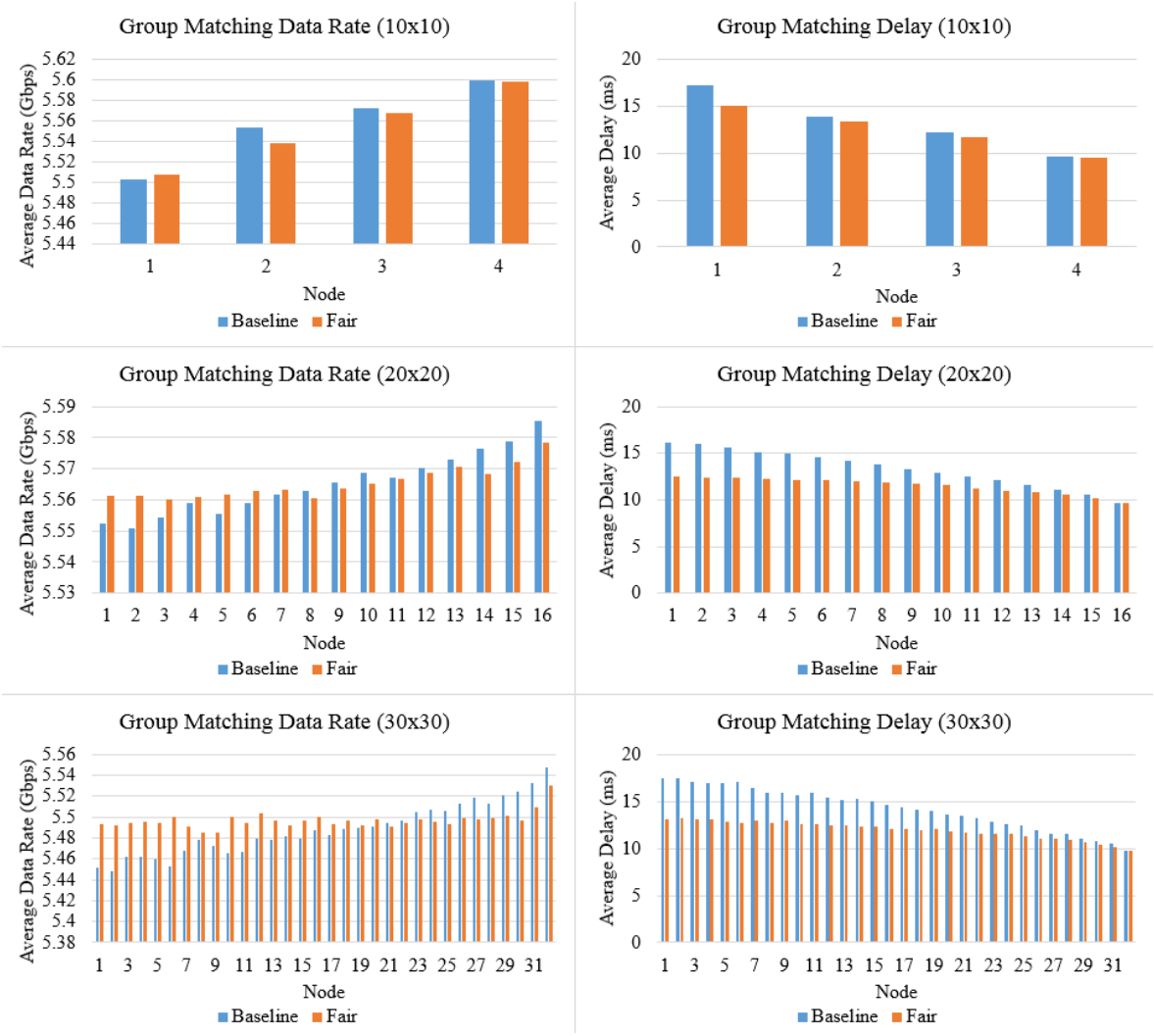


Figures 89 (top left), 90 (top right), 91 (middle left), 92 (middle right), 93 (bottom left), and 94 (bottom right). The average data rate and delay are shown for each experiment in which stable matching fails to produce perfect matches in at least one timestep.

Matching Algorithm	Standard Deviation		
	10x10	20x20	30x30
Baseline	0.03453	0.01276	0.03704
Stable	0.03562	0.00602	0.00945
Fair	0.03089	0.00499	0.00889
Percent Reduction	10.5385	60.8673	76.0048

Table 11. Standard deviation of data rates achieved by baseline, stable, and fair stable matching algorithms. The percent reduction is calculated with (baseline – fair) / baseline.

Fair stable matching works similarly to fair greedy matching except pairing stability is maintained among nodes that are equally disconnected. Pairing stability improves connectivity.



Figures 95 (top left), 96 (top right), 97 (middle left), 98 (middle right), 99 (bottom left), and 100 (bottom right). The average data rate and delay are shown for each experiment in which group matching fails to produce perfect matches in at least one timestep.

Matching Algorithm	Standard Deviation		
	10x10	20x20	30x30
Baseline	0.03513	0.00969	0.02472
Stable	0.02973	0.00515	0.00796
Fair	0.03344	0.00491	0.00760
Percent Reduction	4.7996	49.3343	69.2374

Table 12. Standard deviation of data rates achieved by baseline, stable, and fair group matching algorithms. The percent reduction is calculated with (baseline – fair) / baseline.

Fair group matching improves fair stable matching by explicitly using group information to prevent nodes that have been frequently disconnected from stealing well established high capacity links. This works better when groups are larger in bigger playgrounds.

5.6 Discussion

Matching Algorithm	Average Data Rate		Average Delay	
	Disconnect	Overall	Disconnect	Overall
Fair Greedy	5.5518	5.5965	12.4424	9.9272
Fair Maximal	5.5522	5.5970	12.3758	9.9000
Fair Stable	5.5555	5.5966	12.2611	9.9193
Fair Group	5.5527	5.5965	12.3881	9.9190

Table 13. Average data rate and delay in 10x10m playgrounds.

Matching Algorithm	Average Data Rate		Average Delay	
	Disconnect	Overall	Disconnect	Overall
Fair Greedy	5.5635	5.5725	11.6742	10.7831
Fair Maximal	5.5605	5.5726	11.8266	10.7777
Fair Stable	5.5644	5.5724	11.5979	10.7704
Fair Group	5.5654	5.5733	11.5111	10.6987

Table 14. Average data rate and delay in 20x20m playgrounds.

Matching Algorithm	Average Data Rate		Average Delay	
	Disconnect	Overall	Disconnect	Overall
Fair Greedy	5.4869	5.4920	12.2342	11.4648
Fair Maximal	5.4868	5.4921	12.3520	11.4638
Fair Stable	5.4891	5.4923	12.0794	11.4106
Fair Group	5.4968	5.5000	11.9627	11.2960

Table 15. Average data rate and delay in 30x30m playgrounds.

Fair group matching produces the fairest networks, highest data rates, and lowest delays in 20x20m and 30x30m playgrounds. In 10x10m playgrounds, fair stable matching achieves the best results in all aspects. Stable greedy and maximal matching outperform fair greedy and maximal matching in larger playgrounds because the most frequently disconnected nodes compete against each other for the same relays. The fair algorithms recurrently form suboptimal pairs and matching failures increase. The connectivity issues of competing nodes are compounded, thereby exacerbating connectivity fairness. Stable group matching distributes connectivity more fairly than fair group matching in 10x10m playgrounds because in-group relaying is prioritized even when other users have been more frequently disconnected.

All fair algorithms reduce the maximum number of timesteps a node is disconnected from baseline algorithms. The percent reductions increase as the number of users and playground size increase. The performance among fair matching algorithms is consistent with the performance among stable matching algorithms, and the actual results are quite similar despite

prioritizing different heuristics in matching. The degree of blockage a node experiences is dependent on node height and FOV. Short nodes have the poorest connectivity in relay VR networks because they often have the smallest FOV. Frequently rerouted nodes tend to be the nodes that are also frequently disconnected.

Stable matching algorithms preserve previously established links and breaks these links in a way that maximizes network stability. Fair matching algorithms are identical in this regard and reap the same benefits as the overall data rates and delays are barely distinct. When there are no matching failures, prioritizing frequently disconnected users accomplishes nothing because there are no disconnected users. The two versions can only be distinguished in scenarios where matching failures occur. In these scenarios, fair matching algorithms greedily maximizes connectivity fairness. The overall net changes in connectivity is close to zero but connectivity more fairly distributed among users. Fair algorithms are more suitable for relay VR networks than purely stable algorithms because the former have better guarantees in the worst case scenario. The frequency of matching failures is shown in table 16. Fair matching algorithms are prevalent for smaller playgrounds.

Matching Algorithm	Average Matching Failures		
	10x10	20x20	30x30
Fair Greedy	3.20%	1.01%	0.39%
Fair Maximal	2.17%	0.69%	0.31%
Fair Stable	3.26%	1.32%	0.53%
Fair Group	3.29%	1.42%	0.54%

Table 16. Percentage of trials with matching failures.

6 Conclusions

Relaying is necessary for dealing with mmWave signal attenuation and blockage in VR networks. Existing relaying schemes frequently reroute nodes and unfairly disconnect shorter nodes. This increases the delay and unevenly distributes QoS among users. Bipartite matching algorithms are modified by adding heuristics to stabilize routing decisions and provide fair connectivity. A novel hierarchical group matching algorithm is proposed to utilize user groups in matching. Network stability is achieved by preserving previously established links. Fair connectivity is guaranteed by prioritizing the most frequently disconnected nodes in matching. The numerical results show that AP height has a significant impact on users QoE. Group matching provides the highest data rates, stable group matching achieves the lowest delays, and fair group matching distributes QoS most evenly to users. Relay VR networks is the solution for enabling multi-user VR playgrounds.

7 References

[1] Omid Abari. 2017. Enabling High-Quality Untethered Virtual Reality. In Proceedings of the 1st ACM Workshop on Millimeter-Wave Networks and Sensing Systems 2017 (mmNets '17). ACM, New York, NY, USA, 49-49. doi: 10.1145/3130242.3131494

[2] M. R. Akdeniz et al., "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1164-1179, June 2014. doi: 10.1109/JSAC.2014.2328154

[3] J. G. Andrews, R. K. Ganti, M. Haenggi, N. Jindal and S. Weber. 2010. A primer on spatial modeling and analysis in wireless networks. In *IEEE Communications Magazine*, vol. 48, no. 11, pp. 156-163, Nov. doi: 10.1109/MCOM.2010.5621983

[4] Christian Bettstetter, Giovanni Resta, and Paolo Santi. 2003. The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. In *IEEE Transactions on Mobile Computing* 2, 3, 257-269, Jul. doi: 10.1109/TMC.2003.1233531

[5] Glencora Borradaile, Phillip Klein, Shay Mozes, Yahav Nussbaum, and Christian Wuff-Nilsen. 2017. Multiple-Source Multiple-Sink Maximum Flow in Directed Planar Graphs in Near-Linear Time. In *SIAM Journal on Computing*, vol. 46, no. 4, pp. 1280-1303. doi:10.1137/15M1042929

[6] Mingzhe Chen, Walid Saad, and Changchuan Yin. 2017. Virtual Reality over Wireless Networks: Quality-of-Service Model and Learning-Based Resource Management. arXiv:1703.04209. Retrieved from https://arxiv.org/abs/1703.04209.

[7] Developer. How to generate nD point process. 2011. Retrieved from https://stats.stackexchange.com/questions/16282/how-to-generate-nd-point-process.

[8] Ford, L. R., and D. R. Fulkerson. 1956. Maximal Flow Through a Network. In *Canadian Journal of Math*. 8, 399-404. doi:10.4153/CJM-1956-045-5

[9] Kevin D. Gale and L. S. Shapely. 1962. College Admissions and the Stability of Marriage. In The American Monthly. 69, 1, 9-15. doi: 10.2307/2312726

[10] Xiaohu Ge, Linghui Pan, Qiang Li, Guoqiang Mao, and Song Tu. 2017. Multipath Cooperative Communications Networks for Augmented and Virtual Reality Transmission. In *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2345-2358, Oct. doi: 10.1109/TMM.2017.2733461

[11] Younghwan Go, Muhammad Jamshed, YoungGyoun Moon, Changho Hwang, and KyoungSoo Park. 2017. APUNet: revitalizing GPU as packet processing accelerator. In Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation (NSDI'17). USENIX Association, Berkeley, CA, USA, 83-96.

[12] HTC Vive. 2018. Retrieved from www.vive.com.

[13] Haitham Hassanieh, Omid Abari, Micahel Rodreguez, Mohammad Abdelghany, Dina Katabi, and Piotr Indyk. 2017. Agile Millimeter Wave Networks with Provable Guarantees. arXiv:1706.06935. Retrieved from https://arxiv.org/abs/1706.06935.

[14] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. 1999. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems* (MSWiM '99). ACM, New York, NY, USA, 53-60. doi: 10.1145/313237.313248

[15] IEEE 802.11ay. 2018. Retrieved from www.ieee.org./11.

[16] Ran Ju, Jun He, Fengxin Sun, Jin Li, Feng Li, Jirong Zhu, and Lei Han. 2017. Ultra Wide View Based Panoramic VR Streaming. In *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network* (VR/AR Network '17). ACM, New York, NY, USA, 19-23. doi: 10.1145/3097895.3097899

[17] Jon Kleinberg and Eva Tardos. 2005. Algorithm Design (1st. Ed.). Pearson/Addison-Wesley, Boston, MA.

[18] Simone Mangiante, Guenter Klas, Amit Navon, Zhuang GuanHua, Ju Ran, and Marco Dias Silva. 2017. VR is on the Edge: How to Deliver 360° Videos in Mobile Networks. In *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network* (VR/AR Network '17). ACM, New York, NY, USA, 30-35. doi: 10.1145/3097895.3097901

[19] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia and J. C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi [Invited Paper]," in IEEE Communications Magazine, vol. 52, no. 12, pp. 132-141, December 2014. doi: 10.1109/MCOM.2014.6979964

[20] Occulus Rift. 2018. Retrieved from www.occulus.com/rift.

[21] Wei Peng, Min Li, Yuzhou Li, Wei Gao, and Tao Jiang. 2017. Ultra-Dense Heterogeneous Relay Networks: A Non-Uniform Traffic Hotspot Case. In *IEEE Network*, vol. 31, no. 4, pp. 22-27, July-August. doi: 10.1109/MNET.2017.1600295

[22] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2017. WiFi-Assisted 60 GHz Wireless Networks. In Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17). ACM, New York, NY, USA, 28-41. doi: https://doi.org/10.1145/3117811.3117817

[23] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parmesh Ramanathan. 2015. 60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling. *In Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS '15). ACM, New York, NY, USA, 71-84. doi: 10.1145/2745844.2745858.

[24] Sanjib Sur, Xinyu Zhang, Parmesh Ramanathan, and Ranveer Chandra. 2016. BeamSpy: enabling robust 60 GHz links under blockage. In Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation (NSDI'16). USENIX Association, Berkeley, CA, USA, 193-206.

[25] TPCAST announces TPCAST 2.0. 2018. Retrieved from www.tpcastvr.com.

[26] Teng Wei and Xinyu Zhang. 2017. Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (MobiCom '17). ACM, New York, NY, USA, 42-55. doi: https://doi.org/10.1145/3117811.3117832

[27] Teng Wei, Anfu Zhou, and Xinyu Zhang. 2017. Facilitating robust 60 GHz network deployment by sensing ambient reflectors. In Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation (NSDI'17). USENIX Association, Berkeley, CA, USA, 213-226.

[28] Wi-Fi Alliance. 2018. Retrieved from www. wi-fi.org.

[29] Yuzhe Xu, Hossein Shokri-Ghadikolaei, and Carlo Fischione. 2016. Distributed Association and Relaying with Fairness in Millimeter Wave Networks. In *IEEE Trans. Wireless. Comm.* 15, 12, 7955-7970, Dec. doi: 10.1109/TWC.2016.2609919

[30] H. Zhang, J. Wang, T. Lu and T. A. Gulliver, "Capacity of 60 GHz wireless communication systems over Ricean fading channels," Proceedings of 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, 2011, pp. 437-440. doi: 10.1109/PACRIM.2011.6032933

[31] Yu Zeng, Honglin Hu, Tianheng Xu and Boqi Jia. 2017. User Pairing Stability in D2D-Relay Networks. In *IEEE Communications Letters*, vol. 21, no. 10, pp. 2278-2281, Oct. doi: 10.1109/LCOMM.2017.2721364

[32] Ruiguang Zhong, Manni Wang, Zijian Chen, Luyang Liu, Yunxin Liu, Jiansong Zhang, Lintao Zhang, and Thomas Moscibroda. 2017. On Building a Programmable Wireless High-Quality Virtual Reality System Using Commodity Hardware. In Proceedings of the 8th Asia-Pacific Workshop on Systems (APSys '17). ACM, New York, NY, USA, Article 7, 7 pages. DOI: https://doi.org/10.1145/3124680.3124723

[33] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y. Zhao, and Haitao Zheng. 2014. Demystifying 60GHz outdoor picocells. In *Proceedings of the 20th annual international conference on Mobile computing and networking* (MobiCom '14). ACM, New York, NY, USA, 5-16. doi:10.1145/2639108.2639121

Appendix A: Stable Greedy Depth 2 Matching

In the case where all nodes v in $\mathbf{V_t}$ can't be matched, a new bipartite graph is formed from nodes in set $\mathbf{V_t}$ consisting of the set of matched nodes $\mathbf{V_t}' \subseteq \mathbf{V_t}$ and the set of unmatched nodes $\mathbf{V_t}'' \subseteq \mathbf{V_t}$, where $\mathbf{V_t}' \cup \mathbf{V_t}'' = \mathbf{V_t}$ and $\mathbf{V_t}' \cap \mathbf{V_t}'' = \emptyset$**.** Depth 2 matching performs another round of matching between $\mathbf{V_t}'$ and $\mathbf{V_t}''$. The output is included in $\mathbf{S_t}$, and is referred to as depth 2 matching. Greedy depth 2 matching consists of two rounds of greedy matching. The first round assigns free parents to blocked nodes, while the second round assigns paired blocked nodes to unpaired blocked nodes.

ALGORITHM A: Stable Greedy Depth 2 Matching

1: **For each** node w **in W**: //Match previous pairs if LOS requirements are satisfied

2: **If** LOS$(w, p_{w,t-1})$ and w has a path to AP with length ≤ 2:

3: $p_{w,t} = p_{w,t-1}$;

4: **For** each node v **in** $\mathbf{V_t}$ s.t. $p_{v,t} = \emptyset$:

5: **If** $\exists u$ s.t. LOS(u, AP), LOS(v, u), $p_{u,t} = \emptyset$, and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$: //Find free depth 1 parent

6: $p_{v,t} = u$;

7: **Else if** $\exists u$ s.t. LOS(v, u), $p_{u,t} = j$ for $j \in \boldsymbol{W - V_t}$ and $p_{j,t} = \emptyset$, and $p_{i,t} \neq u$ for $\forall i \in \boldsymbol{W}$: //Find free depth 2 parent

8: $p_{v,t} = u$;

9: **Else if** $\exists u$ s.t. LOS(u, AP), LOS(v, u), $p_{u,t} = j$ for $j \in \boldsymbol{W}$, and $p_{j,t} = i$ for $i \in \boldsymbol{W - V_t}$: //Replace leaf node

10: $p_{v,t} = p_{u,t}, p_{u,t} = \emptyset$, goto 4;

11: **Else if** $\exists u$ s.t. LOS(u, AP), $p_{u,t} = j$ for $j \in \boldsymbol{W - V_t}$, $p_{i,t} = u$ for $i \in \boldsymbol{W}$, LOS(v, $p_{u,t}$) and $p_{i,t} \neq p_{i,t-1}$:
//Replace depth 2 parent whose child was paired in this timestep

12: $p_{v,t} = p_{u,t}, p_{u,t} = \emptyset$, goto 4;

13: **Else if** $\exists u$ s.t. LOS(u, AP) , $p_{u,t} = j$ for $j \in \boldsymbol{W - V_t}$, $p_{i,t} = u$ for $i \in \boldsymbol{W}$, and LOS(v, $p_{u,t}$): //Replace depth 2 parent

14: $p_{v,t} = p_{u,t}, p_{u,t} = \emptyset$, goto 4;

The algorithm first assigns parents to their child from the previous timestep in lines 1-3. When a relay node communicating on LOS with the AP becomes blocked, all its descendants default back to LOS communication with the AP if they are not blocked. This is because the AP cannot use the same route to transmit packets to these descendants. Network latency and overhead is minimized if the descendants switch back to LOS communication with the AP. These nodes will also be able to act as relays for other nodes.

Blocked nodes and descendants are assigned parents in lines 4-14. The algorithm minimizes rerouting, and network latency and overhead by first assigning free parents to blocked nodes in line 5-6. The algorithm then searches for free depth 2 parents in line 7. If all candidate parents are in depth 3 branches, then it becomes necessary to break established links. This process is detailed in figure A. Since detaching a leaf breaks only the leaf node's link, leaf

nodes with LOS to the AP are first assigned to blocked nodes in lines 9-10. Once the leaf is detached and assigned, two depth-2 branches are formed. It is possible that the leaves of these branches can be assigned to a node that could not be matched in an earlier pass. The algorithm performs additional passes over unmatched blocked nodes using goto 4. Depth 2 nodes with LOS to the AP are then traversed in lines 11-14. These nodes have blocked children that have either been paired from the previous timestep or the current timestep. Pairings formed in the current timestep are altered first in order to preserve established links. Depth 2 nodes are forced back to LOS communication with the AP, and their parents are assigned to the blocked node. This results in two depth-2 branches that can each accommodate a blocked node where goto 4 is used again.

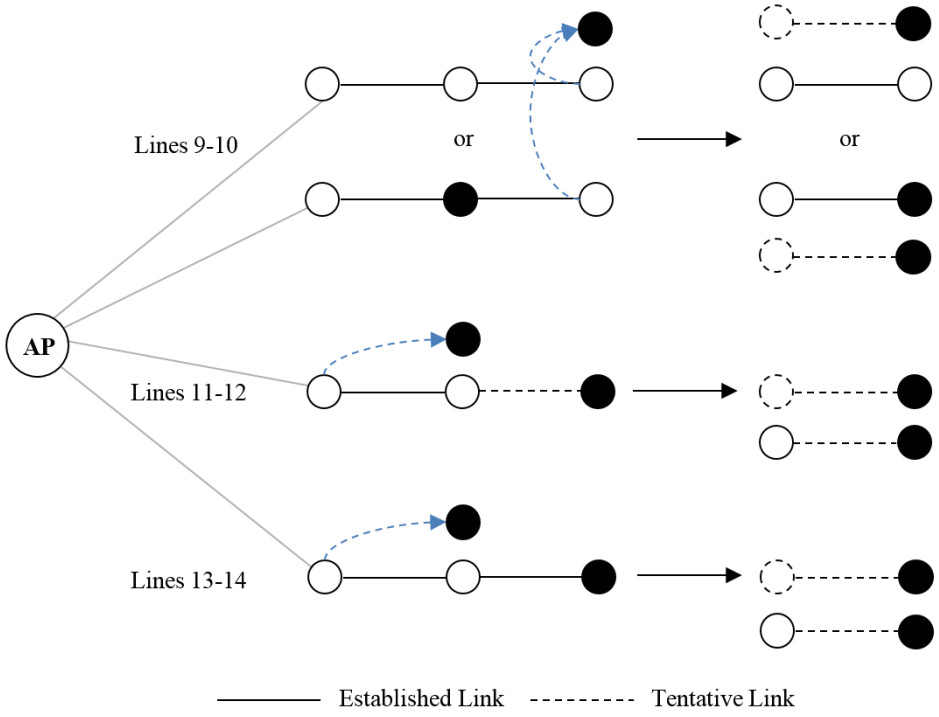


Figure A. Greedy depth 2 matching priority for breaking established links. The figure illustrates which links are replaced first starting from line 9 in algorithm A. Solid lines represent pairings from previous timesteps while dotted lines represent pairings made in the current timestep. Empty circles represent relay nodes; filled circles represent blocked nodes. A dotted circle indicates that rerouting has occurred for that node in the current timestep.

Appendix B: Stable Matching Preference

The stability values of stable Gale-Shapley matching algorithm with different preferences are shown in Figure B. Rerouting frequency decreases with group size when the AP is placed at 2m, but remains fairly consistent when the AP is placed at 4m. This is because blockage occurs densely when the AP is placed at 2m. As group size increases, it more likely for nodes within the same group to be paired. Pairs between nodes of the same group tend to be more stable because nodes in the same group maintain their proximity to one another. The number of reliable candidates therefore increases with group size and network stability is improved. Stability remains fairly the same when the AP is placed at 4m regardless of group size because blockage occurs sparsely. Short nodes are often blocked, and tall candidate relays tend to be far away and part of another group. As group size increases, the number of nodes in close proximity increases, making it more like that a group member will block the LOS to the tall relay node. The probability of blockage increases with group size and network stability suffers.

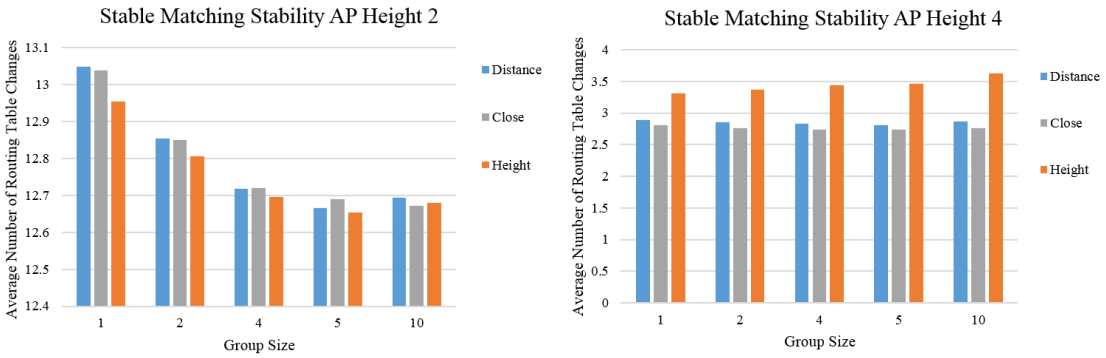


Figure B. Stability of stable Gale-Shapley matching with different preferences. Left: Rerouting frequency when the AP is placed at 2m height. Right: Rerouting frequency when the AP is placed at 4m height. 100 nodes are simulated on a 50x50m playground. Distance refers to preferring the nearest node, height refers to preferring the tallest node, and close refers to preferring the closest node that is taller than the node itself. Parents prefer the nearest node for all cases.

Distant links are more prone to blockage than shorter links because the number of nodes that can move to block the LOS between distant pairs is greater than that of neighboring pairs. Preferring height is advantageous at 2m AP height because there are many blocked nodes and tall relays are highly contested. The nearest blocked node is paired due to parent's preference. Preferring height exhibits inferior performance at 4m AP height because blocked nodes are often paired with distant relays.

Close stable matching slightly outperforms distance stable matching in both AP scenarios for all group sizes. This is because the nearest relay may be short, making it more likely to be blocked. Thus, a taller relay node is more appropriate. When the AP is placed at 4m, height stable matching achieves the best balance between selecting a tall relay and a close relay; whereas, when the AP is placed at 2m, close stable matching achieves the best balance.

Appendix C: Connectivity at 0m AP Height

The performance of each matching algorithm when the AP is placed at 0 height is shown below.

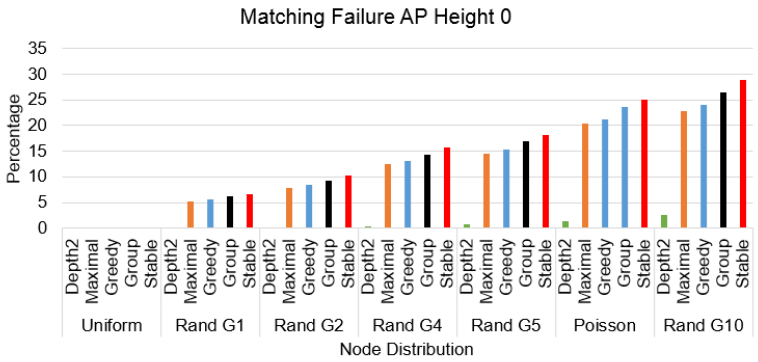


Figure C. Matching performance when the AP is placed on the floor. 100 nodes are simulated on a 50x50m playgrounds.

Greedy and maximal matching have similar performance because the majority of cases in which they fail are ill-posed: the number of blocked nodes is smaller than the number of nodes relays. In well-posed problems, maximal matching is successful in all cases. It is easy to arrive at a perfect match because there are many connected users in the graph that can reroute transmission to a node. Group and stable matching have the lowest connectivity since they impose more constraints on the matching result. Stable matching forces all matching pairs to have no incentive to undermine their assignment by joint action, whereas group matching only enforces stability among group members.

Greedy depth 2 matching succeeds in 97.32% of scenarios where nodes are distributed in groups of 10. Node connectivity cannot be completely guaranteed, but overall VR quality should still be enjoyable. Depth 2 matching improves connectivity at the cost of heavier workloads for user nodes. Although application quality might not be hindered, more power is consumed. Since user nodes are resource-constrained, depth 2 matching is less practical for relay VR networks. On the other hand, maximal matching fails with 22.821% probability, and matching failures are detrimental to system performance. APs should be placed at vantage points to accommodate resource-constrained user nodes. This setup requirement is effortless compared to setup required by other systems, i.e. the placement of multiple mmWave mirrors in [1] would be tedious in multi-user scenarios. Matching performance can be improved by allocating AP-only space. This is discussed in the next appendix.

Appendix D: Impact of AP Space on Connectivity

Allocating AP space improves connectivity. Nodes are generated further away from the AP, so they take up less of the AP's FOV. This allows more nodes to communicate on LOS with the AP. The average matching failure of each algorithm over all distributions with varying AP-only space is shown in figure D.

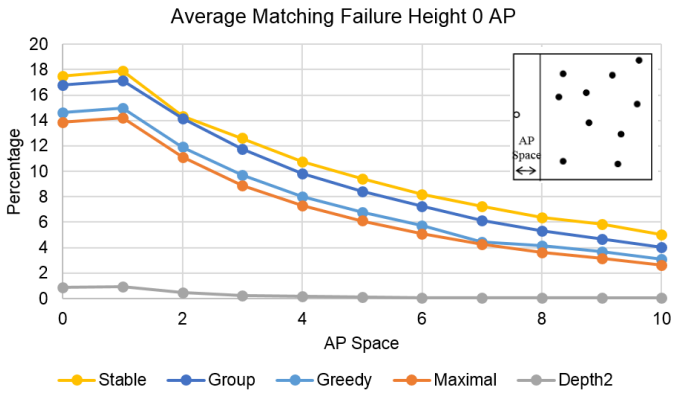


Figure D. The average matching failure over all distributions of each algorithm for different amounts of AP-space is shown when the AP is placed on the floor. AP-space is a rectangular space defined by its width as shown in the top-right diagram. 100 users are simulated on a 50x50m playground.

The average matching failure increases at 1m because nodes can no longer be generated directly to the side of the AP. Nodes are confined to a smaller space while AP FOV is not improved. Matching performance starts to improve at 2m of AP-space. The improvement decreases as more space is allocated because nodes are generated in smaller spaces. This results in more congestion and blockage. Allocating more than 10m of AP-space improves matching performance, but users' space becomes more limited. Since user experience is affected by the size of the playground, we consider only up to 10m of AP-space for our system. Greedy depth 2 matching improves to 99.968% and maximal matching improves to 92.969% at 10m of AP-space. If VRHMDs can tolerate heavier workloads while maintaining application quality and AP space can be allocated, then the AP can be placed on the floor. Otherwise, user connectivity cannot be guaranteed if the AP is placed on the floor.

Appendix E: Edge Computing

Edge computing has been shown to provide latency improvements for computationally intensive mobile applications compared to cloud-offloading models. It is also more energy efficient for mobile devices to offload rather than perform local processing. In some particular scenarios, application performance is enhanced when edge servers are accessible through Wifi. It is important to characterize scenarios and computations that benefit from edge computing for enabling interactive mobile video applications. Computer vision applications have a wide variety of computational complexity. These computations are offloaded to determine whether edge computing can enable compute intensive HD video applications.

Experiments are performed with a Huawei P9 model that is connected to an intel i7-4770 3.4 GHz edge server via an in-lab Wifi network. Offloading is handled by Sapphire, which runs on Apache Harmony RMI, and ported BoofCV android samples are used.

Only two of the applications experimented with are shown here. They suffice to illustrate the general effects of edge computing on mobile computer vision applications.

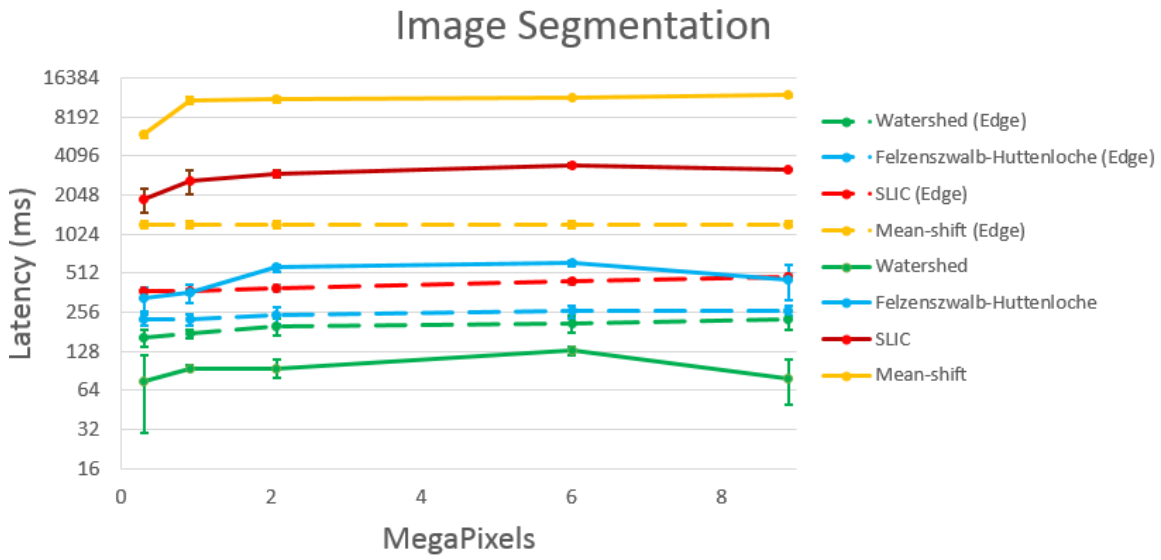


Figure E. shows the latency of various image segmentation algorithms with varying image size. The dotted lines represent the latency per image when offloading, and the solid lines represent the latency for local processing.

Figure E confirms edge computing can achieve lower latency relative to local processing. Only local processing of the watershed algorithm outperforms edge computing. The computational latencies indicate that it is not yet possible to perform real-time image segmentation in video even at the lowest (340x480) resolution.

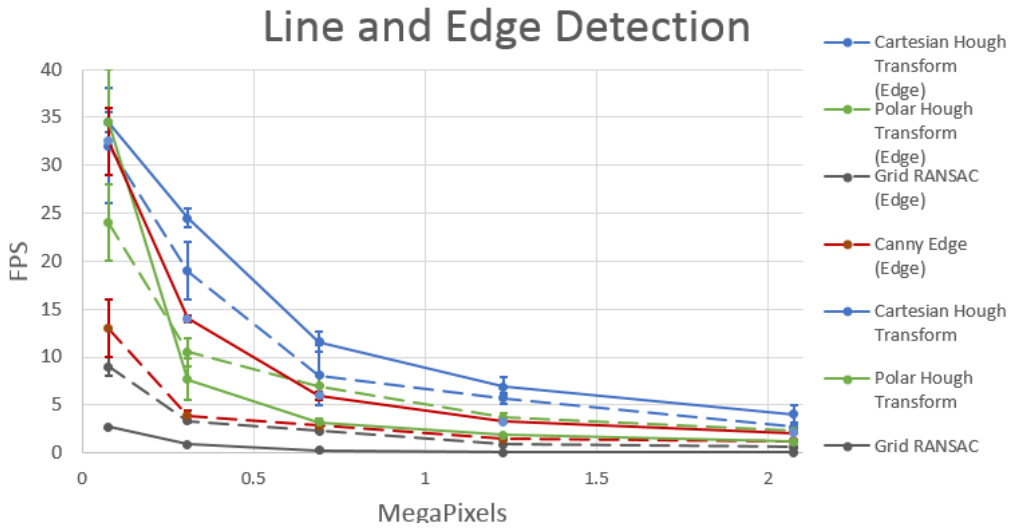


Figure EE. shows the latency of line and edge detection algorithms. Polar Hough transform and Grid RANSAC benefit from computational offload, while other algorithm results in performance degradation.

Figure EE shows that it is not yet possible to enable any real-time computer vision applications at full HD video even with edge computing. The FPS severely decreases as video resolution is increased. This is consistent for every application. The right most data points in figure 2 show the FPS for full HD video (1920x1080). All applications whether offloaded or not fails to display even 10 FPS.

If applications perform too simple computations, the overhead resulting from computational offload will outweigh any performance enhancements from edge computing. On the other hand, if applications perform too complex computations, edge servers will be unable to meet latency requirements for sustaining 30 FPS.

Since algorithms in BoofCV have not been optimized, the offloaded computation cannot fully take advantage of the computation resources available on edge servers. There are also inefficiencies in the android kernel (i.e. double buffering). Furthermore, the Wi-Fi environment has not been optimized to enable high bandwidth for the mobile device. In order to maximize the number of applications that can benefit from edge computing, offloading overheads need to be minimized while achieving maximum performance on edge servers.

The reported measurements show that there is much more room for improvement in order to enable high quality interactive mobile video applications with edge computing.