Simplifying the Emulation of WLAN Environments: Design and Implementation of a Multi-Client WLAN Traffic Emulator

UNIVERSITY OF TURKU
Department of Computing
Master of Science (Tech) Thesis
Cyber Security Engineering
May 2026
Antti Tuominen

Supervisors:
Saku Lindroos (University of Turku)
Petri Sainio (University of Turku)
Jani Turpeinen (Sasken Finland Oy)

UNIVERSITY OF TURKU
Department of Computing

Antti Tuominen: Simplifying the Emulation of WLAN Environments: Design and Implementation of a Multi-Client WLAN Traffic Emulator

Master of Science (Tech) Thesis, 83 p.
Cyber Security Engineering
May 2026

Wireless communication is becoming more popular, especially the usage of Wireless Local Area Networks (WLANs). Although WLAN provides convenient Internet access, the IEEE 802.11 standard family that enables the wireless communication results in variety of diverse WLAN environments. The constant development of new amendments in the IEEE 802.11 standard family offers consumers many technologies for improving the WLAN environment. The constant development and improvement in data rates and multi-client WLAN environments requires a solution for running benchmark tests for WLAN environments.
The requirement for creating real-world WLAN environments is necessary when benchmark testing the newly constructed WLAN environments. Therefore, emulation is the best solution because it focuses on replicating real-world scenarios with physical devices and software rather than utilizing available simulation software. Especially multi-client emulation is a great way of creating WLAN environments. There are commercial products available for multi-client emulation, which have limitations in deployability and upgradability.
This thesis proposes a multi-client WLAN emulator that can be upgraded to support the latest amendments of the 802.11 standard family. To simplify the emulation of WLAN environments, a Python script is created for controlling the proposed multi-client emulator setup. The Python script creates a possibility for the multi-client emulator setup to utilize multiple connected Wireless Network Interface Cards (WNICs) for generating WLAN traffic with iPerf2 tool. The proposed multi-client emulator is developed in a way that it can manage the newest amendments of the 802.11 standard family, while also supporting all three 802.11 frequency bands.
The proposed multi-client emulator setup was evaluated against data collected from physical WLAN devices. Various WLAN environments were emulated with the multi-client emulator to prove the versatility of the tool. The tests run on the multi-client emulator were compared to results gathered from real-world client devices such as mobile phones and laptops. To compare the results of the multi-client emulator to the results of the real-world devices, iPerf2 data of the multi-client emulator and real-world client devices was collected for analysis. The comparison shows that the proposed multi-client emulator can be used to emulate real-world WLAN environments, which makes it a suitable option for commercial WLAN traffic emulators.

Keywords: WLAN, multi-client emulation, IEEE 802.11, wireless network interfaces

Contents

List of Figures

List of Tables

Listings

1 Introduction

Wireless Local Area Networking (WLAN) technology deployment has increased rapidly in the last two decades. It is necessary that the technologies evolve since the devices that require connectivity are evolving too and require more bandwidth and higher data rate to work as intended. The economic value of WLAN devices has been growing rapidly, increasing 150% between 2018 and 2025 [1]. This alone is enough to understand that WLAN technology is one of the most important ways for users to connect to the Internet. Therefore, it is necessary to have the best possible connections all over the world. Whether at home or in a public environment, users are likely to connect wirelessly to an Access Point (AP).

To have best possible connection in every situation, testing the WLAN network is a suitable way to make sure the environment is adequate for supporting multiple clients and high data rates. Testing the WLAN network is important for ensuring the proper performance of the WLAN environment. The true value of the benchmarking comes when the results of benchmarks are compared to existing systems and their respective measures [2].

1.1 WLAN client emulator

In this thesis WLAN multi-client emulator is proposed, where Wireless Network Interface Cards (WNIC) can be connected to an AP while generating real life network traffic. This kind of wireless traffic generation can become time consuming and complicated if many devices such as mobile phones or laptops need to be configured separately. The process can be simplified by connecting multiple WNICs to a single host device that will then simulate traffic through the connected WNICs controlled by a script. This type of multi-client emulator enables the emulation of multiple clients more easily and more realistically than conventional WLAN simulation software. This thesis proposes the implementation for a multi-client network traffic emulator for WLAN environments. The proposed system is designed to be sufficiently flexible to allow testing variable WLAN environments. In addition to flexibility, it is important to consider compatibility with future WLAN standards and devices, which creates the requirement for the WLAN traffic emulation system to support the latest and future WLAN technologies. Therefore, the software code should be kept simple and readable to facilitate future maintenance and updates. Keeping the system and code simple provides flexibility and helps the development further.

It is also important to note that the WNICs used in this thesis can vary from future implementations. The WNICs do not necessarily need to be external USB powered WNICs, but they are suitable for emulating WLAN traffic. Keeping in mind that computers have differences and they utilize different hardware for WLAN connectivity, therefore the script needs to be suitable no matter what the hardware solution is.

The goal of this thesis is to implement system for WLAN traffic emulation with possibility to utilize multi-client emulator using multiple WNICs. The main challenges involve creating a system that is both cost effective and sufficiently versatile to generate and transmit suitable data for specific benchmark tests. Achieving this requires a clearly defined system design, carefully selected requirements, and a well-defined objective to guide the development process. Testing the system is part of the objective, and by generating necessary benchmarks for variable environments it is possible to achieve promising results of the proposed multi-client emulator. The multi-client emulator proposed in this thesis provides a useful basis for testing and selecting an appropriate AP for a given WLAN environment.

1.2 Research problem

While commercially available wireless network emulators are widely available, licenses for the software can be expensive since commercial products often provide more options for traffic generation and traffic analysis. Therefore, the goal of this thesis is to improve on commercial solutions and provide a cost-effective alternative that can be adjusted according to the user's environment. Since existing emulators often provide more advanced hardware and software and may be subject to paid licenses, a simple script for emulating WLAN environment can offer a useful alternative.

Based on this, the research questions set for this thesis focus on how to develop a more cost-effective and precise tool for emulating WLAN traffic. The research questions are:

1. What hardware is required to simplify WLAN emulation without having to gather multiple physical client devices?
2. How to connect multiple WNICs to a single AP for emulating a multi-client WLAN environment?
3. What kind of traffic can be sent with the emulated multi-client WLAN environment?

1.3 Thesis structure

The rest of the thesis is structured as follows. Chapter 2 provides the background of WLAN technologies, delving into the history and status of the current IEEE 802.11 WLAN standards. Chapter 3 discusses how to benchmark test wireless environment and what kind of studies can be found similar to this thesis. Moreover, chapter 3 focuses on the existing commercial network emulation solutions and what their capabilities are. Chapter 4 presents the proposed multi-client WLAN traffic emulation system, including the overall system setup as well as the hardware and software components used in its implementation. Chapter 4 describes the proposed system as precisely as possible to enable the possibility for future improvements. System test results and the collected data are presented in Chapter 5. Finally, Chapter 6 summarizes the research findings and outlines possible directions for future improvement.

2 Background

In this section, an overview of the 802.11 based standards is provided. To provide idea on how IEEE 802.11 environments work and what enables the use of WLANs, it is necessary to cover the topic of Local Area Network (LAN) as a background. Typical example of LAN is a basic home Ethernet network in which devices are connected to a consumer-grade router via twisted pair cables. The IEEE 802.3 Ethernet standard is the base for Ethernet communication to work, where amendments are proposed for the standard for improving the communication [3]. To make the communication work the network requires public Internet Protocol (IP) address which is assigned by Internet Service Provider (ISP), connecting the LAN to the global Internet. This LAN has range of local IP addresses that are distributed to connected client devices by DHCP server that is running on a router that hosts the LAN.

For large enterprises and organizations LANs become increasingly complex and require a well-designed structure and increased bandwidths. Bandwidth requirements for modern LANs have increased substantially due to applications such as video and voice requiring increasingly more bandwidth [4]. While media applications do require higher data rates and large bandwidth, new applications like Virtual Reality (VR) or Augmented Reality (AR), Industrial Internet of Things (IIoT) and even gaming benefits from higher data rates and large bandwidth [5].

While wired LANs can provide the necessary data rate and bandwidth for these applications, it does not have the simplicity a WLAN can provide. While WLAN has its challenges when enabling connection for the more advanced applications, the development of the IEEE 802.11 standard has improved significantly since the original standard's release in 1997 [5].

2.1 WLAN communication

The Institute of Electrical and Electronics Engineers (IEEE) have created standardizations for various wireless networking technologies of which the 802.11 WLAN standard will be the focus of this thesis. As the 802.11 standard is hard to commercialize the Wi-Fi Alliance (WFA) created the Wi-Fi brand for the IEEE 802.11 standard. Each amendment that will be covered in this thesis has a corresponding version provided. Table 2.1 shows the corresponding versions and standards that are in use.

IEEE Standard	**Wi-Fi version**	**Year**
IEEE 802.11b	-	1999
IEEE 802.11a	-	1999
IEEE 802.11g	-	2003
IEEE 802.11n	Wi-Fi 4	2009
IEEE 802.11ac	Wi-Fi 5	2013
IEEE 802.11ax	Wi-Fi 6/6E	2021
IEEE 802.11be	Wi-Fi 7	2024

Table 2.1: IEEE 802.11 wireless standards corresponding Wi-Fi versions [5].

While wired LAN is connected through cables, it gives the opportunity to extend the network with wireless routers or APs. Wireless devices communicate through radio waves making communication less reliable compared to wired networking. Still, the demand for WLAN communication is increasing continuously. One reason for this is the simplicity of WLAN deployment, as only a single AP connected to the Internet via an Ethernet cable is required.

WLAN APs available in the consumer market provide support for variable 802.11 standards. While some APs provide support for IEEE 802.11ac, some for IEEE 802.11ax and even the latest IEEE 802.11be standard. Each standard provides capabilities aimed at improving WLAN communication. For instance, IEEE 802.11 standards provide support for different modulation techniques, these are for example Orthogonal Frequency-Division Multiplexing (OFDM) or Orthogonal Frequency-Division Multiple Access (OFDMA). In addition to variable modulation techniques they can have multi-user connections like Multi User Multiple-Input Multiple-Output (MU-MIMO).

Depending on the standards supported by the AP, it can utilize three different frequency bands 2.4GHz, 5GHz and 6GHz bands. From these, the 2.4GHz and part of the 5GHz bands are unlicensed and allocated to Industrial, Scientific and Medical (ISM) applications [5]. In addition, part of the 5GHz and 6GHz bands have wider bandwidths and more channels in the shared but unlicensed bands assigned in the Unlicensed National Information Infrastructure (U-NII) bands [5]. Dual-band technology is a way to utilize two frequency bands in a way that congestion can be avoided in the network by changing the used band automatically. With new standards, the 6GHz band provides the possibility to support tri-band communications. It is also important to note that while the frequency band increases, the range of the AP decreases. This means that even if the 6GHz band can support higher speeds, it is not necessarily the best option for certain scenarios. The modulation and multi-band technologies introduced in the IEEE 802.11 standards are covered in detail in sections 2.2, 2.3, 2.4, 2.5 and 2.6.

As there are variable technologies used in WLAN communication, it is also important to have methods to test that all the required capabilities work. Although some of the discussed communication techniques cannot be tested with the hardware available for this thesis, a discussion of the technologies is provided to understand the advancements of WLAN environments.

2.2 IEEE 802.11

The IEEE 802.11 standard defines the communication mechanisms between APs and client devices in wireless local area networks, particularly at the Physical (PHY) layer and the Medium Access Control (MAC) sublayer of the data link layer related to the lower layers of the Open Systems Interconnection (OSI) model [5]. The PHY and MAC layer protocols ensure robustness and interoperability between devices supporting different standards [5].

As physical layer is the lowest of the OSI model, it decides what protocols to use to send raw data between AP and client device. The WLAN standards use variety of PHY layer specifications with the target of aggregating the throughput of the connection, the main PHY layer techniques included in the original 802.11 standard are Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) [6]. The MAC layer, on the other hand, handles addressing and access control, ensuring that data frames are delivered to the appropriate destination. In this the client device scans the availability of a channel on the connected AP and sends the data if it is available, if not the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) comes in to prevent collisions of frames during transmission from happening [6]. There are two schemes that MAC layer uses to ensure the correct channels and addresses, these are Distributed Coordination Function (DCF) and Point Coordination Function (PCF), both can be used, although PCF is not commonly used anymore due to its poor robustness against hidden nodes [7]. With 802.11e standard two new MAC techniques were introduced [6]. These methods were Hybrid Coordination Function (HCF) which is based on the PCF and Enhanced Distributed Coordination Function (EDCF) which is based on DCF. With 802.11e standard there were also two enhancements to the Acknowledgement (ACK) frames, where only one ACK is needed for block of frames instead of one ACK for each frame [6]. The enhancement in the ACK frame improved the throughput together with the introduction of other enhancement such as direct link with two client devices in a network [6].

To be able to start communicating AP and client device need to form association. The way that the client can find the AP is that the AP sends broadcast frames, advertising its capabilities to client devices. Before the Extensible Authentication Protocol Over LAN (EAPOL) authentication process, the client device sends association request to the AP to which the AP answers with an association response. The association request and association response frames enable the client and the AP to exchange information on their capabilities and supported standard features, thereby allowing the connection requirements and parameters to be adjusted accordingly. After association, the client device and AP exchange EAPOL authentication frames to start the data transfers.

The data transfer has many variables that affect how the data is handled, whether it is high or low priority data. The IEEE 802.11e introduced Quality of Service (QoS) enhancements to MAC protocol by enabling voice and video transport with QoS [8]. To prioritize traffic, QoS defines priority levels for 802.11 frames. The MAC will utilize different technical functions if the QoS is utilized in the 802.11 communication standard [9]. Enhanced Distributed Channel Access (EDCA) technique is one of the functions, and it provides the possibility to process packets differently depending on the class of the packet and channel capacities [8]. With EDCA it is possible to specify the priorities and categories for QoS as shown in table 2.2.

Access category	**Priority level**
Background	1
	2
Best effort	0
	3
Video	4
	5
Voice	6
	7

Table 2.2: Priority levels in QoS [8].

The IEEE 802.11 standards have a long history, with the first standards released in 1997, although WLAN devices were not yet widely available in the consumer market at that time. All the previously discussed modulation, ACK knowledgement and association capabilities have been developed and enhanced for a long time now. The original 802.11a, 802.11b and 802.11g amendments are considered as legacy technologies, thus they fall out of scope of this thesis, and will not be included in the following discussion. The first relevant standard for this thesis is IEEE 802.11n. Each standard from 802.11n to 802.11be have some improvements over the generations before and each standard has interesting capabilities which will be covered next.

2.3 IEEE 802.11n

As the standards introduced before the IEEE 802.11n were capable of 54Mbps data rate, the main purpose of the 802.11n amendment was to double the data rate to 100Mbps [10]. To achieve this, modifications to both PHY and MAC layers were necessary for achieving the improved data rates for 802.11n. When the task group was established to develop the 802.11n amendment in 2003, many proposals for the 802.11n amendment were declined, and in the end two proposals from World Wide Spectrum Efficiency (WWiSE) and TGn Sync group were combined. The joint proposal of the two groups formed the basis of the 802.11n amendment in 2006 [11].

The 802.11n amendment provides High Throughput (HT) between devices as the 802.11n-2009 amendment states that it can support at least the 100Mbps as it was designed to do [12]. The 802.11n amendment contained significant improvements. The proposed improvements focused on the PHY and MAC layers, since those are the layers that control the wireless communications.

2.3.1 IEEE 802.11n PHY features

Modulation and Coding Schemes (MCS) and HT Protocol Data Unit (HT-PPDU) are features that are integrated in HT devices [12]. The differences in the PHY layer compared to previous standards are MIMO, Spatial Multiplexing (SM), Space-Time Block Coding (STBC), Low Density Parity Check (LDPC) encoding and Antenna Selection (ASEL). 20MHz and 40MHz channels are compatible with this amendment and the PPDUs can be transmitted in HT or non-HT stations [12].

To achieve higher data rates and greater bandwidth, the 802.11n amendment introduced MIMO, which uses multiple transmit and receive antennas [11]. A MIMO configuration can, for example, use a 2x2 antenna setup, meaning that the device has two transmit antennas and two receive antennas. To create data streams, the antennas have certain coding techniques like STBC which uses for example Almouti coding, which can encode one spatial stream into two space-time streams [11]. One of the main features of the PHY layer in the 802.11n amendment is transmit beamforming, where spatial matrices are found by using channel estimates, which improves the receiving signal by focusing on prevalent modes of transmission for channel [11].

2.3.2 IEEE 802.11n MAC features

In addition to the PHY features of the 802.11n amendment, there are also new MAC layer features introduced. Among the most significant new MAC features are frame aggregation, Block ACK, Power Save Multi Poll (PSMP) operation, Reverse Direction (RD) features and protection mechanisms that supports even the non-HT devices [12]. Support for delay sensitive standards was introduced which emphasizes improvements on MAC layer in QoS [11]. Other significant features introduced in 802.11n include the addition of Voice over WLAN (VoWLAN) and Voice over IP (VoIP).

In 802.11n amendment there are two types of frame aggregation in the MAC layer: MAC Protocol Data Unit Aggregation (A-MPDU) and MAC Service Data Unit Aggregation (A-MSDU). The MPDU and MSDU differ from each other in the sense that A-MSDU is at the top of the MAC layer and A-MPDU is at the bottom of the MAC layer [10]. The two aggregation methods differ in the robustness of the aggregation methods. While A-MSDU fails if the frames have bit errors, the A-MPDU does not, and the receiver can continue to next MPDU according to the header signature [10].

The 40MHz channel was introduced with 802.11n amendment through channel bonding, which combines two 20MHz channels into one 40MHz channel [13]. One of the challenges the channel bonding introduces is managing both 20MHz and 40MHz capable devices. With the dual-band technology and use of 2.4GHz and 5GHz frequencies, the managing of the devices, and especially neighboring APs becomes challenging due to increased traffic from both frequency bands [10]. The management of neighboring APs is important when designing WLAN environment, therefore it is required for the APs to scan for neighboring Basic Service Sets (BSS) before it can enable 40MHz BSS. This scan can prevent enabling channel bonding if there are overlapping channels during the scan of BSS [10].

Channel utilization is one issue resolved in the 802.11n amendment, because the number of devices connecting to the network can cause congestion. The Power Save Multi-Poll (PSMP) was introduced to improve power consumption and channel utilization when small amounts of packets are sent periodically [10]. Compared to the legacy amendments, the 802.11n amendment had many improvements. The demand for WLAN devices increased significantly after the introduction of 802.11n, which increased the requirement for higher data rate. Therefore, the development of new generation of 802.11 amendment began.

2.4 IEEE 802.11ac

While the number of devices that communicate wirelessly grows, so does the traffic. Streaming videos and playing games become more popular and newer and faster standards are required for better connectivity. IEEE 802.11ac amendment, which is improvement from the previous 802.11n amendment, seems to be the solution for this. The aim of the 802.11ac amendment was to have support for 500Mbps for single link, 1Gbps support for multi-station throughput, focus on the 5GHz band and backward compatibility to legacy devices on the 5GHz band [14].

Compared to the 802.11n amendment that was capable of 100Mbps speed, the 802.11ac amendment is more ambitious and has more improvements compared to the 802.11n and legacy amendments. In addition, 802.11ac amendment also improved some security features to have more robust and confidential connections and communication. The 802.11ac amendment aims to have Very High Throughput (VHT) capabilities, to improve from the previous 802.11n amendment in a way that devices supporting the VHT capabilities are also capable of HT communication [15]. One of the main features that was introduced in the 802.11ac amendment is Downlink Multi-User MIMO (DL-MU-MIMO), which enables AP to create four A-MPDUs, and each MPDU is transmitted simultaneously to one associated client [15]. When designing WLAN environments, it should be noted that devices compliant with IEEE 802.11n support only HT features and cannot utilize VHT features.

2.4.1 IEEE 802.11ac PHY features

The PHY layer in 802.11ac amendment has many improvements compared to 802.11n amendment. These features include channel bonding two 40MHz channels into 80MHz channel being mandatory with optional support for 160MHz and 80+80MHz channel width [15]. Other significant features are mainly optional, but one mandatory feature is support for Single-User (SU) PPDUs, and the optional features are sounding protocol supporting beamforming, Multi-User (MU) PPDUs and support for MCS 8 and 9 [15].

Compared to the 802.11n amendment, where 40MHz channel was largest supported channel bandwidth and supported only one or two spatial streams, the 802.11ac amendment mandates only one spatial stream with 80MHz channel. One spatial stream reduces the system overhead since more antennas mean less complicated implementations [16]. Data rates are also increased with support for the MCS 8 and 9. These MCS indexes are based on 256 Quadrature Amplitude Modulation (QAM), with this the improvement in data rate compared to 802.11n which is based on 64-QAM is up to 33% [16].

With the introduction of the MIMO in the 802.11n, the 802.11ac improves this feature by adding the Multi User (MU) possibility. This allows the AP to communicate with up to four client devices at the same time [14]. For optimal performance of the MU-MIMO the AP should have more antennas than the total number of spatial streams, and the AP requires channel state information from client devices to form antenna weights [14]. MU-MIMO technology improves the multi-client WLAN environments, which introduces the importance of benchmarking the WLAN environment with multi-client emulation.

2.4.2 IEEE 802.11ac MAC feature

In the MAC layer, there are also some mandatory features such as support for A-MPDU padding of VHT PPDU, support for VHT single MPDU and support for responding to bandwidth indication in device that does not support HT and HT duplicate Request to Send (RTS) frame [15]. Optional 802.11ac MAC features include, for example, advanced methods for MPDUs with higher octet count and VHT link adaption for utilizing the MU-MIMO with modulation schemes [15].

802.11ac combines the features of A-MPDU and A-MSDU presented in the 802.11n amendment to achieve better efficiency [14]. Overlapping in channels that have larger width than 40MHz becomes inevitable because of the channels getting wider. At least with the neighboring BSS's, avoiding the overlapping is hard in the 80MHz or wider channels [14]. To address this, the 802.11ac amendment improves the co-channel operation with enhancements that focus on avoiding the overlapping on channels [14].

2.5 IEEE 802.11ax

After the 802.11ac amendment the development continued, and the next amendment called IEEE 802.11ax was introduced in 2021 [5]. Unlike 802.11ac, the 802.11ax amendment improves operations on both 2.4GHz and 5GHz bands, but in addition to that support for operations in the 6GHz frequency band was introduced. This enables the congestion on the lower bands to be moved in the higher 6GHz band, which is not as popular as the other bands. In addition, with the introduction of the 6GHz band the 802.11ax devices can support tri-band operation, which relieves the congestion on the 2.4 and 5GHz bands. To increase the throughput of communication, the 802.11ax amendment focuses on the efficiency of communication [17]. Therefore, 802.11ax amendment does not provide similar increase in data rates as

802.11ac.

The devices that support the 802.11ax amendment are capable of High Efficiency (HE) communications between the devices. To be compatible with the previous standards, there are many requirements specified for each frequency band. Once again, the new amendment brings new features to the PHY and MAC layer to improve from previous amendments. All these new features make it possible for the amendment to support larger bandwidth and higher throughput. It is also important to note that in commercial Wi-Fi the Wi-Fi 6 means only the 2.4 and 5GHz bands, and the Wi-Fi 6E includes all three bands.

2.5.1 IEEE 802.11ax PHY features

With the 802.11ax amendment, the previously optional features such as support for OFDMA, DL MU-MIMO by AP, and DL MU-MIMO reception for client device were made mandatory. To enable beamforming for a beamformee, the HE sounding protocol was introduced in the 802.11ax amendment. [18] In addition to the mandatory features in the PHY layer, there are also few optional features. These optional features include support for HE-MCSs 10 and 11, UL MU-MIMO and preamble puncturing [18].

The new features in the PHY layer are the multi-user antennas and the possibility of OFDMA communication. This allows the client device to communicate with AP by having less interference and more efficiency. With simultaneous use of OFDMA and MU-MIMO, the traffic is handled efficiently. While OFDMA is good for small packets, and makes the transporting easier for small packets, the MU-MIMO makes sure that the large packets are transferred efficiently.

2.5.2 IEEE 802.11ax MAC features

The new MAC features introduced by this amendment can be divided into AP-specific and client device-specific features, as well as into mandatory and optional features. The mandatory features for APs include support for role of Operating Mode Indication (OMI) and support for individual Target Wake Time (TWT) operation, which are optional for client devices in 802.11ax amendment. Client devices received couple mandatory features in 802.11ax amendment. To make the communication between client devices and APs more efficient, the client devices require support for two Network Allocation Vector (NAV) operations and support for multiple BSSID operations. Some optional features introduced in the amendment that support multi-client environments are broadcast TWT, UL OFDMA-based Random Access (UORA), and multi-TID A-MPDU operation. [18]

Achieving the higher throughput and other advantages introduced by the 802.11ax amendment required the implementation of several new features. Not only do the new features affect the data rate and the ways to affect a consumer directly, but in this amendment, efficiency is also considered. While TWT helps the devices to keep the line clear when there are no connections required, it can also have impact on the devices lifetime which makes the environmental impact decrease. TWT is an effective technology in congested WLAN environments since the devices only send data when required, which improves the multi-client communications in WLAN environment.

2.6 IEEE 802.11be

IEEE 802.11be is the latest amendment introduced as part of the 802.11 standard. The 802.11be amendment brings improvements to multi-client communication and introduces new features. The 802.11be amendment was introduced in 2024 and it has a theoretical data rate of 23 Gbps [5]. For this amendment, the devices are capable of Extremely High Throughput (EHT) features [19]. This amendment again supports tri-band technology operating on all three available frequency bands.

The 802.11be amendment has been in development for a long time, beginning in 2018 when topic of EHT raised interest [20]. Due to 802.11be being introduced in 2024, the amendment has not yet seen widespread adoption. The 802.11be amendment provides significant improvements over the 802.11ax amendment. One of the most important upgrades compared to previous amendment are support for 320MHz channels, 16 spatial streams, 4k-QAM and Multi-Link Operation (MLO) [20]. Again, as the data rate increases for 802.11be amendment, the possibilities of the connections improve. This enables the usability for wireless communication to be used in advanced medical tools and for example more advanced audio and video streaming [20].

2.6.1 IEEE 802.11be PHY features

For the 802.11be amendment, there are many new features and improvements in the PHY layer to provide a higher peak data rate. The main improvements are multi-RU possibility for single client, new modulation technique with 4096-QAM and enhanced preamble designs [21]. These improvements to the 802.11be amendment enable better connectivity between devices and possibility to create WLAN environment that can, for example, support high quality video streaming.

While the channel bonding in the previous 802.11ax amendment was capable of 160MHz, the 802.11be can utilize the 6GHz band to get 320MHz bandwidth. The 320MHz bandwidth can be achieved by being contiguous and located only on 6GHz band or noncontiguous by existing partly on 5GHz and partly on 6GHz band [21]. The noncontiguous method introduces the possibility to utilize bandwidth of 160MHz on the 5GHz band and at the same time 160MHz on 6GHz, which combines the bandwidth to 320MHz total [21]. With new improvements in bandwidth, some changes to the Resource Unit (RU) allocation have been made in the 802.11be amendment. One of the main improvements made to RU allocation is the possibility to utilize multiple RUs for one client device [21].

To simplify the multi-RU combination schemes the 802.11be amendment has split the RU allocations into small and large RU types [21]. The small and large RU types have different combinations that are allowed, depending on bandwidth and RU-tone combination. The small part of the RU allocation focuses on smaller tones and bandwidths reach maximum of 80MHz [21]. While this is made to simplify the combinations, the large RU types have a lot more combination possibilities and they focus from 80 to 320 MHz bandwidths. The RU-tones included in the large RU types are above the maximum of 106-tones in the small RU types, and they reach up to the newest 3 X 996-tone introduced in 802.11be amendment [21]. The improvements provided in the multi-RU compared to the previous RU techniques seem to focus more on the larger bandwidths and support those because they are better choice for higher data rates.

The higher modulation scheme in the 802.11be amendment improves the peak data rate achieved with the 802.11be amendment. With the 4096-QAM modulation scheme the transmission rate can increase 20% compared to the previous amendment, and this is with the same coding rate [21]. With the new and improved PHY layer techniques the 802.11be amendment has the possibility to reach the desired peak data rate.

2.6.2 IEEE 802.11be MAC features

One of the main features that 802.11be amendment has introduced is the MLO. This technology is interesting, especially when multi-client connections are in use, since it can divide the traffic simultaneously on each 2.4, 5 and 6GHz bands. To achieve the utilization of all the spectrum resources, there are new technologies introduced, including new spectrum management, coordination and transmission mechanisms over the three bands [21]. The MLO option in an AP creates a separate network that client devices can connect to, only requirement is that those devices need to have the support for the MLO technology.

Multi-Link Devices (MLD) have differences in the capabilities that it can operate on. Since some MLDs have multi-radio capabilities and some have only single-radio capabilities, there are different types of MLO Operations for both single- and multi-radio technologies [22]. To have enhancement in the throughput of single-radio MLD, the 802.11be amendment has defined a method to enable most of the concurrent dual-radio benefits. This MLO operation is called Enhanced Multi-Link Single-Radio (EMLSR) mode, and it creates a possibility for single-radio MLD to listen to at least two links at the same time [22].

While Multi-Link Multi-Radio (MLMR) capable devices have the possibility to transmit and receive simultaneously over multiple links, the simultaneous transmit and receive can have some issues. These issues are caused by In-Device Coexistence (IDC) interference, which can happen if device has multiple radios to transmit and receive simultaneously [22]. To avoid this type of issue the 802.11be amendment has introduced Simultaneous Transmit and Receive (STR) and Nonsimultaneous Transmit and Receive (NSTR) operations [22]. One more MLO operation that was introduce in 802.11be is Enhanced MLMR (EMLMR), which introduces possibility to dynamically reconfigure spatial multiplexing on each radio [22].

In this case with the new MLO operations the 802.11be amendment also requires some enhancements for multi-link channel access. In the 802.11be amendment the new channel access rules are required for the EMLSR, NSTR and EMLMR operations [22]. In both EMLSR and EMLMR operations, the client receives a control frame from the AP on one of the links. Based on this frame the client switches its operation to that link and continues transmission and reception using the corresponding radio resources [22]. This type of operation that EMLSR and EMLMR operations offer cannot work with the NSTR operation since it does not allow simultaneous transmission and reception on links. Therefore, 802.11be amendment has additional rule for the NSTR that aligns end time of PPDUs transmitted to NSTR client at the same time [22].

Traffic Identifier (TID)-to-link mapping is one of the new features introduced in 802.11be amendment to enhance the MLO and help to identify traffic types. This TID-to-link mapping mechanism helps in separating the traffic and prioritizing critical and important traffic, since it is possible to assign subset of TIDs to certain dedicated links [22]. The TID-to-link mapping can be tied closely to QoS and how the different QoS values are assigned to different links.

2.7 Takeaways for WLAN multi-client emulation

As discussed in this chapter, the IEEE 802.11 standard has developed significantly, and new amendments have introduced faster peak data rates and more reliable communication methods. Even though the newest 802.11be amendment introduces suitable data rates and reliability, the development of IEEE 802.11bn amendment is already ongoing. Ultra-High Reliability (UHR) Study group has produced some information on the new amendment, L. Galati-Giordano et al. provide more insight into the 802.11bn amendment in [23].

The 802.11 standard and its amendments offer a lot of options for designing and configuring WLAN environments, which makes the need for benchmarking WLAN environments beneficial. As this chapter introduced new technologies within the 802.11 standard family, especially the ones supporting multi-client scenarios, the emulation of WLAN environment becomes more complicated. It is important to know how the devices supporting different standards work and if they are suitable for specific WLAN environments. Therefore, benchmarking the different WLAN environments, by using multi-client emulators, is useful for figuring out what is the most suitable solution for certain environments.

3 Existing WLAN emulators

As new technologies and standards advance, new and suitable solutions for WLAN environments are required. There are many things to consider when creating a benchmarking environment for figuring out which WLAN standard would be best suitable. As new amendments are published, benchmarking WLAN environments need to evolve at the same time. The constant cycle of introducing new IEEE 802.11 amendments creates a loop where new communication technologies and operations are introduced, and they need to be tested, which requires the constant evolvement of the testing equipment and technology.

Creating a simulation of WLAN environments is one possibility to test different operations introduced in 802.11 amendments, some simulation tools are for example GNS3, Cisco Packet Tracer, Cisco VIRL and ns-3. While simulating the WLAN environments might be easy and cost efficient, the simulations do not represent actual physical WLAN environments. This is because when the WLAN environment is simulated with the available simulation tools, the variables affecting real-world WLAN environments are not necessarily taken into consideration. These variables are, for example, devices outside of the WLAN environment sending traffic over the air, which could interfere with the WLAN environment. These need to be taken into consideration when benchmark testing WLAN environment, and therefore emulating the network seems to be better solution.

While devices are sending traffic to each other in physical WLAN environments, there will be faulty packets and other variables that will influence the connection. Therefore, to test real-life scenarios by using a simulation tool might not be enough because the physical environment is not necessarily taken into consideration. Emulation on the other hand, creating real-world scenarios is considered one of the most versatile options for testing overall end-user experiences [24].

Some existing solutions have been proposed to improve the emulation of WLAN environments. The existing WLAN emulation solutions differ in their implementation, while others try to create emulation through virtual machines, others are creating multi-client scenarios with physical WNICs connected to a computer. Both approaches, whether implemented using virtual machines or physical devices, have their respective strengths and limitations. The idea is mainly the same for each existing solution, creating environments as close to a real-world environment as possible to have reliable results of WLAN environments.

Traffic generation is a big part of the network emulation [24]. Without the traffic the emulated environment would generate very little traffic that would allow needed information to be gathered. Therefore, both the reviewed research and this thesis focus on what kind of hardware is needed to generate sufficient traffic to emulate real-world WLAN environment.

3.1 Emulators proposed in academic research

One of the earliest research studies presenting a multi-client 802.11 WLAN emulation by G. Alvarez et al. [25] mimicked a classroom environment where many laptops connect to an AP and generated wireless network traffic. According to the authors it might be the first emulation focused on performance testing the IEEE 802.11-based networks. The focus of the research was to emulate the classroom environment, so that even the WNICs used in the tests are same as the ones in the laptops provided in the Plan Ceibal education premises [25]. For this kind of high-density classroom scenario, it is important that same WNICs are used in the emulator for creating environment that is as close to real-world scenario as possible [25].

The test results presented by Alvarez et al. in [25] are compared to real traffic that laptops send in a classroom. The emulator setup created for the study can achieve similar results in throughput and airtime as the laptops. This establishes a strong base for emulating high density scenarios with a Linux machine sending wireless traffic from many WNICs. While the paper was published in 2016, the IEEE 802.11 standards were not as capable as today, therefore the WNICs used in the paper were 802.11n devices. As the used test devices supported a now out dated standard the presented solution would not be the best solution to benchmark IEEE 802.11-based wireless environments today.

The same emulator setup presented by Alvarez et al. was also used by G. Capdehourat et al. in [26]. They both focus on creating multi-client emulation setups designed for the same high density classroom setup. In [26], the software tools are discussed in more detail, and it introduced some useful information on what tools exist in helping to generate the wireless traffic. The iPerf tool utilized by Capdehourat et al. in [26] will be discussed more in detail in section 4.2.1. In addition to iPerf, Capdehourat et al. utilized wget, PhantomJS and Pytomo [26]. These tools are used to test the emulator with different type of data traffic, for example with wget, it is possible to send HTTP traffic while Pytomo tries to emulate YouTube video traffic [26]. Out of the three tools wget and PhantomJS are similar, one downside of wget is the inability to support JavaScript, therefore PhantomJS is used to support it [26]. The goal of the paper was to create a multi-client emulator, which according to the test results worked fine, but the Pytomo tool supported only one client at a time [26].

Another research done regarding wireless network emulation is presented by T. Kawai et al. in [27]. Kawai et al. had few goals to accomplish in their paper, where they tried to create a setup that was low cost, reproducible, had easy preparation and managed to use native programs. The setup was created using already existing simulators and virtual machine environments, which were used to add additional nodes to the setup. The created system would have a host that runs the network simulator, which would be able to manage traffic to virtual WLAN devices [27]. The prototype proposed by Kawai et al. emulates high fidelity environment. The network simulator and virtual machines used for prototypes can be different, but for their research Kawai et al. used Scenargie 1.8 for network simulator and QEMU 2.1.2 for virtual machines.

Emulating physical devices with virtual machines is an interesting approach to emulating WLAN environment. At least from the results that was provided in [27], the prototype emulates the real-world environment well. Since the virtual machines are running on one host, that will create some issues in the setup. One of the main issues with virtualization discussed in [27] is the lack of CPU cores, which limits the amount of virtual WNICs to use in the prototype. While technology has improved since Kawai et al. published their research in 2017, most likely if the setup was recreated today, there might be possibility to create more virtual WNICs for different scenarios.

In [28] R. Liu and N. Choi created a multi-client test scenario. The presented system focused on analyzing capabilities of IEEE 802.11ax standard, and more precisely focuses on OFDMA and TWT capability analysis. The setup Liu and Choi created contains total of 10 clients which include desktops, laptops and other IEEE 802.11ax standard supporting client devices such as smart phones. The proposed system with multiple physical devices, however, requires additional maintenance, because there are 10 different clients and each needs to be managed on their own respective User Interface (UI). Solution for simplifying the maintenance of the setup has automated parts where the parameters for devices are configured automatically, this solution is discussed in more detail in [28].

As the focus of Liu and Choi in [28] is to test if the IEEE 802.11ax standard is faster, the results focus more on capabilities of the IEEE 802.11ax standard. The presented results included, for instance, a test of power consumption of IEEE 802.11ax supporting device, which has increased power consumption compared to previous IEEE 802.11 standards [28]. While the applications are run directly from client device, that creates the actual real application wireless traffic. This can help when the tests are specifically done for testing certain 802.11 standards, but when comparing this to the emulation systems created in [25], [26] and [27] the mobility of the setup is complicated. But as pointed out, that is not the goal of the setup in [28].

In 2023 research was published by V. Planchart et al. to test Ultra-Dense Wireless Networks (UDN) [29]. The goal of the research was to create an environment that could emulate the UDN, and traffic in it, but having low costs in the process. The system presented by Planchart et al. utilized Raspberry Pi's to create a multi-client environment. All the Raspberry Pi's were controlled with a computer through Secure Shell (SSH) connection, and the setup was divided into two networks, control and test networks [29]. The setup seems easy and has one control interface for each client on the test network, which makes it easier to send network traffic to the environment with iPerf.

The tests were however conducted in an 802.11n based WLAN environment [29]. Planchart et al. managed to produce multiple tests, and the result seems to have been achieved as planned. The setup is simple and easy to integrate into different environments, depending on the capabilities of available hardware. Planchart et al. do not however discuss if the emulator would work with newer 802.11 standards, which would have been good basis for future emulation of newest IEEE 802.11 standard environments.

One of the most capable open access network emulators that is publicly available to the research community is Colosseum presented by L. Bonati et al. in [30]. Colosseum is one of the most used wireless networking emulators, which can create scenarios to test RF, Cellular, IEEE 802.11-based communication and even Unmanned Aerial Vehicles (UAV) [30]. Colosseum can be accessed remotely to create and test real-world WLAN environments or new technologies with capable hardware. It has 128 Standard Radio Nodes (SRNs) [30], which indicates that it can emulate large number of wireless clients including WLAN devices. Colosseum is also capable of creating more complex test scenarios and producing corresponding results. For example, the paper presents a Signal to Interference plus Noise Ratio (SINR) test scenario involving moving clients.

In addition to the academic research done on wireless network emulators, a wide variety of literature exists focusing on different kinds of WLAN performance analysis. For example, some performance evaluations can be found that have similar test setup as the emulators. One of the newest performance evaluations done for IEEE 802.11be is presented by H. Mäntylä in [31]. Some research for WLAN traffic emulation was also done in 2001 by M. Kojo et al. [32], although the research did not specifically focus on IEEE 802.11 standards and therefore was left out of thorough discussion.

The most frequent goal in the existing research was creating low-cost and easy to configure and transferable wireless networking emulators. These have their limitations depending on the hardware. When the idea is to create low-cost emulator, the result cannot possibly have every possible scenario for testing capabilities of all 802.11-based standards. Therefore, some organizations offer more capable emulation setups to have the latest test setups with latest possible technology available. These setups on the other hand come with higher prices and with limits depending on the providers' support and software.

3.2 Existing commercial wireless networking emulators

Many companies are providing wireless network emulators for testing Devices Under Test (DUT). Whether the DUT is client or AP the idea is that the emulator can manage creating scenarios for testing the DUT. Transmit (Tx) and Receive (Rx) operations are important to test for the RF layer [33]. The main goal of the wireless networking emulators is to create real-world network conditions for testing purposes.

When comparing the commercial wireless network emulators to the solutions proposed in recent academic research presented in section 3.1 and freely available network emulators, it is clear to see which is more capable of producing more versatile environment. There are many commercial emulators available, and each have their own technologies for testing. The organizations discussed in this section are offering the newest emulators for testing the most recent IEEE 802.11be standard. In addition, all the presented emulators are compatible with older IEEE 802.11 standards, but the organizations offer also older versions of their emulators. This section presents the differences in commercial products and tries to compare them to the existing research in wireless network emulation.

3.2.1 Spirent TestCenter

Spirent Communications is one of the organizations that provides a wireless network emulator. Spirent Communications was acquired by Keysight Technologies in October of 2025 to enhance network automation and satellite emulation [34]. The acquisition happened during the writing of this thesis, and therefore Spirent and Keysight are kept apart from each other, and their solutions are discussed separately.

From Spirent, one wireless client emulator was found that could be compared to the solution in this thesis. Spirent TestCenter is a client emulator for emulating large number of 802.11 a/b/g/n/ac clients [35]. The Spirent TestCenter has many radios controlled by NIC, with that and the test modules included the product offers many possibilities for WLAN emulation. The clients can operate on 2.4 and 5 GHz channels while it is also possible to emulate different protocol modes and spatial streams for real world client scenarios [35].

With the Spirent TestCenter, there are many possibilities for testing how the IEEE 802.11-based devices work under the scenarios emulated by the TestCenter. While the research discussed in section 3.1 focus on testing the throughput and how the DUT handles a lot of traffic. With the Spirent TestCenter, there are more possibilities to test other than the throughput. Multi-client scenarios can be created with the Spirent TestCenter, which is especially important for testing the MIMO and OFDMA capabilities. In addition, the TestCenter can do security type testing, mixed standard client testing, maximum client testing and association processing with different authentication selections [35].

The Spirent TestCenter can also be used as a WLAN traffic sniffer [35]. This provides a real-time packet list display, to analyze the packets further and see how the DUT responds to created environment. In general, this part is important and the commercial products all provide some type of log view mode. While more affordable systems created in the research field might not directly support the logging of the packets, it is simple to setup on another machine to sniff the traffic.

3.2.2 Keysight

Keysight offers many solutions for wireless networking purposes and especially for wireless network emulators [36]. Their solutions do not only focus on the IEEE 802.11-based device testing, but they have wireless network emulators that can help in testing Long Term Evolution (LTE) and 5G connections or devices. Keysight also provides chambers combined with wireless network emulators, to create suitable environments for Over-The-Air (OTA) testing [36]. For wireless network emulators that focus on the IEEE 802.11-based devices, Keysight provides WaveTest products for IEEE 802.11ax testing. In addition, they have emulator that is capable of IEEE 802.11be testing purposes.

WaveTest 6

WaveTest 6 is a wireless network emulator that focuses on the IEEE 802.11ax devices with 2.4 and 5GHz bands. As discussed in section 2.5, the IEEE 802.11ax standard introduced multi-client features, where dense environments got a performance gain compared to earlier standards. The WaveTest 6 emulator is aimed to test the multi-client scenario in IEEE 802.11ax environments, and more specifically it is designed to simulate multi-user OFDMA [34].

The main purpose of the WaveTest 6 is to validate APs, by creating real-world scenarios, where many clients are sending traffic to an AP [34]. There is possibility of creating a scenario where clients are different distances from an AP. The emulator consists of baseband, custom radio and FPGA technology, with the custom hardware it can emulate 500 clients and even 2000 clients in multi-port mode [34]. With the WaveTest 6 emulator, the user has possibility to simulate clients, generate traffic and analyze it with multi-user OFDMA sniffing capability. The WaveTest 6 emulator is already capable of creating multi-client scenarios, but as new IEEE 802.11-based standards are introduced, new emulators are required.

WaveTest 6E

As the IEEE 802.11ax standard adopted the 6GHz band, the WaveTest 6 needed an upgrade. Therefore, new WaveTest 6E was provided by Keysight. The new WaveTest 6E wireless network emulator has focused on the same aspects as the predecessor WaveTest 6, at least the multi-client OFDMA is one of the main features advertised on both emulators [34], [37]. Similarly, the newer WaveTest 6E also supports up to 500 emulated clients [37].

In the WaveTest 6E documentation there is a mention about "Golden Client", which can be used to test large 802.11 infrastructure networks [37]. This is the term that is used for possibility of creating 500 separate clients per wireless interface. With this, it is possible to analyze the highest limits of an AP to see how it acts in dense multi-client environments [37]. For network environments, the multi-client testing possibility is extremely important, as the IEEE 802.11-based standards are extremely popular today. The testing of, for example, an AP is important, but each test also needs to be repeatable. Therefore, the WaveTest 6E documentation mentions the high accuracy for client generation to reproduce same tests reliably [37]. Again, as in WaveTest 6, WaveTest 6E has the options for creating automation and to grant easy access to the emulator.

One interesting addition that is mentioned in the documentation of the WaveTest 6E is possibility to test Ethernet-based networking products [37]. This shows the extent that the commercial products can offer for benchmarking purposes of new technologies. All the testing possibilities are not necessarily required for one's use, but it seems that there is no option to only get the wireless network emulator.

E7515W UXM for Wi-Fi 7 testing

For testing IEEE 802.11be based devices Keysight provides an emulator called E7515 UXM platform. It can perform basic RF, throughput and load testing, but in addition to that the platform manages cellular and IEEE 802.11-based standard interworking tests [38]. The improvement compared to the WaveTest 6 and 6E emulators is based on an existing cellular testbed. This enhances the test setup and improves the IEEE 802.11ax and 802.11be based devices multi-client tests, like the

OFDMA [38].

Although the previously discussed research-based emulators are able to address throughput and jitter metrics, commercial products provide improved performance in these areas. The E7515W UXM platform in addition to the throughput and jitter metrics can conduct packet loss, retransmissions and aggregation [38]. While the platform can test the interworking of cellular and IEEE 802.11-based connections, it enables the possibility for Voice-Over Wi-Fi (VoWiFi) calling [38]. The integrated VeriWave technology enables the testing platform to perform reliable benchmarking and even Quality of Experience (QoE) validation [38].

With the introduction to the 320MHz bandwidth with the 802.11be amendment, the emulators require advancements on that part also. In the E7515W UXM platform, Keysight introduces the 320MHz bandwidth in the 6GHz band range up to 7.125GHz band [38]. One of the main ideas of the emulators is to create a repeatable, easy to use and fast test setup. User interface is provided with the Keysight E7515W UXM platform and with it, repeatability, faster setup of the environment and the automation of the tests are improved [38]. In addition, as in the previous versions of Keysight's emulators, it is possible to do analysis on the system and debug issues and problems that might occur in the environment. The possibility to analyze the test straight from the platform is significant advantage in the commercial products over the more affordable research-based emulators.

3.2.3 Alethea WiCheck

Alethea Communications Technologies is an organization that offers products in wireless testing and measurement areas. They are offering a solution for real-world client emulation, which can emulate up to 1 500 concurrent IEEE 802.11be-based clients [39]. With the WiCheck emulator, it is possible to create many different scenarios and send actual real-world traffic that is offered with the solution. There is a controller that can create so-called golden clients and it can also manage real devices, like laptops and phones in addition to certain radio or accessory testing [39]. With the real-world devices and traffic that are available on the WiCheck emulator, it seems that the WiCheck has a lot to offer, especially when it also supports the newest IEEE 802.11be devices.

More specifically the golden client subsystem in WiCheck provides the possibility to functionality test IEEE 802.11be environments by creating real-world environments [40]. The golden client subsystem has configuration for pre-built test cases for easy access and usability, but there is also possibility to create new and more specific use cases [40]. With this it is easy to create different types of real-world scenarios, whether it is small office spaces or large-scale network environments, the WiCheck seems to provide tools to create various environments. The WiCheck's golden client subsystem offers functionality testing for the IEEE 802.11be based environments. These tests include for example MLO, 320MHz bandwidth and 4096 QAM and multi-band testing with the 2.4, 5 and 6GHz bands [40].

One interesting addition with the WiCheck is the possibility to test the environmental variabilities with the test houses provided with the WiCheck solution [41]. With the test houses solution, it is possible to create an environment that has IoT devices in kitchen and for example some 4K streaming devices in a living room [41]. This offers scalable testing in the most common scenarios that utilize the IEEE 802.11be standard, which is important for validating the most suitable solution for certain environment. Based on all the information found about WiCheck, it seems to be versatile solution for validating and testing IEEE 802.11be environments with automation and simple controller interface to configure the environments.

3.2.4 QA Cafe CDRouter

QA Cafe offers their own solution for network testing, which is known as CDRouter. Automation is big part of the solution that QA Cafe is providing and the tests can be automated to run continuously, 24 hours a day [42]. This is of course important for testing the stability of an environment or AP to create best scenarios or devices possible. With the CDRouter, it is possible to test more than just IEEE 802.11-based environment, for example WAN connectivity and routing functions can be tested with the CDRouter [42].

With the CDRouter it is possible to simulate clients with IPv4 or IPv6 hosts, and the clients can act as devices from 802.11n up to 802.11be standard [42]. CDRouter includes a lot of test cases for different purposes to test performance or security for example. There are many test cases provided with the CDRouter, since it comes with 10,000 loaded test cases [42]. This provides a lot of possibilities for testing network environments, not all the tests are made for wireless environments that utilize IEEE 802.11-based devices, but there are still large number of scenarios to test.

Performance tests in the CDRouter contain 22 modules which have different tests that can be run with either IPv4 or IPv6 hosts. There are a total of 2250 test cases that can be run only to test the performance of an AP for example [43]. There are also multi-client test scenarios, but it is not stated how many clients can be emulated in the tests. Some interesting test cases for the IEEE 802.11-based standard, in addition to the performance test include multiport tests and Wi-Fi range vs rate tests that the CDRouter offers. For the multiport test there are 122 test cases [44] and for the Wi-Fi range vs rate test there are 400 test cases [43]. CDRouter has a lot to offer for network testing, and it is a well-suited tool for automating tests.

3.2.5 Candela technologies

Candela technologies is a company that focuses on creating and supporting network testing and emulation solutions, and they offer many different client emulators that focus on the IEEE 802.11 standard family [45]. From their website, they have many solutions that offer different capabilities for testing purposes. These client emulators are small compared to the previously discussed emulators, for example, the CT521b-2ax4 is portable since it fits in a travel bag [46]. All the solutions offered have different number of clients that can be generated and depending on the supported IEEE 802.11 based standard the capabilities of the emulators can differ [45].

Candela technologies also provides their own LANforge WiFIRE software for use with, for example, the CT521b-2ax4 emulator [46]. With WiFIRE emulation scenarios can be created and from the web page it seems it has a lot of configuration possibilities. Portability seems to be one of the differences compared to the other commercial products, but even with newer versions of the emulators, they seem to be too large to be carried in a bag. Since there are many products, the users can get the most suitable solution for the specific use cases they have.

3.3 Multi-client emulation solutions

As discussed in this chapter, there are many options for emulating WLAN environment, which complicates the choice for best suitable tool. In general, one of the best ways to create actual real-world scenarios would be to gather smart phones, laptops and IoT devices and one at a time connect them to AP and create the environment that way. This task would, however, be complicated since many devices need to be controlled at the same time, and therefore the emulation setups introduced above are important in testing the IEEE 802.11 based environments. In addition to that, the collection of the devices that support the newest standards is expensive and while the WLAN standards are evolving, the devices need to be updated again.

While the academic research discussed in section 3.1 does introduce suitable methods for emulating WLAN environments, the standards that were utilized are not the latest. Therefore, the commercial products are more suitable for testing wireless network. The commercial products are expensive, for example, on the Candela technologies website [45], there are some pricing examples for the emulators. The pricing of course depends on the provider and the capabilities the emulator offers. For one time purchase the commercial products are expensive at least when compared to buying a few new phones and laptops for emulating network environment. Over the long term, commercial products may prove cost-effective, given that they provide continuous support and can be updated in response to the evolution of IEEE 802.11 standards. Commercial products also provide some extra testing possibilities that could appear to be useful in certain scenarios.

The system introduced in this thesis is similar to the ones discussed in section 3.1. This way the outdated versions of client emulators are updated to support newest 802.11 standards. In addition, the goal is to create a solution that is capable of reproducing the same tests and scenarios to make sure same environment can be repeated in future benchmarks. Since the solution is tied together by a control script, it is also possible to easily modify for specific needs, for example make it portable by using it with a laptop, or more capable by using more powerful hardware and stable PC components. It is also possible to manage how many WNICs can be integrated into the system, to create more high-density environments.

4 Design and development of the multi-client emulator system

Chapter 3 described the existing literature and some commercial products that could be used in emulating WLAN environment. The presented commercial products are costly and provide numerous features that are unnecessary for benchmarking WLAN environments. Therefore, this thesis proposes a more cost-effective and finer tuned for users need option for benchmark testing the multi-client WLAN environments. One significant benefit of the system proposed in this thesis is the possibility to use USB powered external WNICs. As USB powered WNICs can be purchased separately, it is possible to connect it to a PC and this way the results of the emulator are more equivalent to real world scenario. Since the setup is controlled by the user, it is ideal to be able to utilize the setup with variable WNICs with different drivers and chips.

There are many possibilities for different kinds of combinations with different devices supporting different 802.11 standards. While the new standards are backwards compatible with the previous standards, the backwards compatibility simplifies the design and creation of the environment. To be able to emulate different IEEE 802.11 WLAN environments, the hardware needs to be selected carefully, especially the WNICs, since the interfaces are the ones deciding how to communicate with a wireless AP. Because modern laptop computers seldom offer multiple USB ports, the USB hubs are also important for the proposed system. There are hubs that support different speeds and to have good enough support for the newer 802.11 standards, the USB hubs need to have support for high-speed devices. The devices and applications used in this thesis will be introduced in more detail in the following sections.

4.1 Hardware

For the hardware part of the emulator, there are many variables to be considered. Since the idea is to create a setup for multi-client emulation, there needs to be at least two WNICs, as well as a laptop for connecting and controlling the WNICs. The setup also requires an additional computer for capturing the generated network traffic. For the system presented in this thesis the server machine will host a server that the WNICs on the multi-client emulator machine can communicate with.

All the WNICs on the multi-client emulator setup are connected to an AP that routes the traffic to correct server address. For the AP it is important that it supports the correct IEEE 802.11 amendments, for the presented setup 802.11be compatible AP is used. Although the server machine and the AP are used to support the testing of the proposed multi-client emulator, the hardware focus of the emulator itself lies primarily on the host machine and the WNICs presented in sections 4.1.3 and 4.1.4.

4.1.1 Server machine

Although this thesis focuses primarily on the host machine, a separate machine hosting the iPerf2 server is also required. A more indepth discussion of the iPerf2 software is provided in section 4.2.1. The server machine is connected to the same AP as the host machine with a wired connection to create an iPerf2 server for the

WNICs on the host machine to communicate with. The PC itself does not need to be powerful for this purpose. For the setup, HP Pavilion 570 PC will be used. It has Ubuntu installed on it and it runs on AMD A8-9600 Radeon R7 CPU with Nvidia GeForce GT 1030 graphics card. In the end the hardware of the server machine does not matter if it can host the iPerf2 server.

The iPerf2 server on the machine is set up so that it runs on the wired network interface of the machine. With iPerf2 it is possible to start one server that can communicate with multiple clients at the same time. This way it is possible to set up different parameters with iPerf2 on the host machine for creating different emulated WLAN environments. In order to emulate Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic at the same time, it is required on the server machine to set up two iPerf2 servers. This can be done on the same interface, so it does not complicate the setup too much. In this thesis some of the tests discussed in chapter 5 require both UDP and TCP servers running on the server machine.

4.1.2 WLAN access point

One important criterion for the WLAN AP chosen for the test setup was for it to support the latest 802.11 standards. For testing the client emulator setup, the Acer Predator T7 access point is used. It can support up to the 802.11be standard, and it has some of the standards new features that were discussed previously in section 2.6. For example, it supports the MLO and 320MHz channels, while offering the capabilities of the 802.11 based standards predecessors.

The Acer Predator T7 AP has simple interface that can be used to control the communication channels. This is one of the reasons the AP was selected for use, since it is simple to activate or deactivate frequency bands. It is also possible to utilize channels that are not congested, to get clean results from the emulator. The

AP allows variable configurations to be used, the configurations for the AP will be discussed more specifically with each test scenario in chapter 5.

4.1.3 Wireless network interface controllers

WNICs are one of the most important parts about the client emulator setup. It is important that they can support many features that the different IEEE 802.11 standards offer. Therefore, it is important to select interfaces with different capabilities. Also, as the latest 802.11 standards enable support for multi-client scenarios, it is recommended to have at least two interfaces for each 802.11 standard to have variability in emulating WLAN environments.

One of the improvements compared to the commercial products would be the transportability of the proposed system, since the commercial products are bigger and most likely connected to a rack. It will be easy to change the environment from one place to another, by just carrying a laptop that acts as the host machine. This simplifies the selection of the interfaces for the emulator. To emulate a client, physical interface is required and since the host machine is laptop, USB powered WNICs were selected.

The WNICs chosen for the proposed system have been listed in table 4.1. As seen from the table, all of the chosen WNICs have been produced by Netgear and each 802.11 standard has two WNICs in use for the emulator. All the selected WNICs are utilized to run tests with the multi-client emulator. Having multiple WNICs supporting different 802.11 standards allow the system to run variable tests for creating different WLAN environments.

Wireless Network Interfaces					
Client	**Model**	**Chip**	**Standard**	**Capabilities**	**pcs**
Netgear AC1200	A6210	MediaTek MT7612u	IEEE 802.11ac	300/867Mbps, dual-band	2
Netgear Nighthawk AXE3000	A8000	MediaTek MT7921u	IEEE 802.11ax	6GHz, tri-band, MU-MIMO, OFDMA	2
Netgear Nighthawk BE6500	A9000	MediaTek MT7925u	IEEE 802.11be	tri-band, MU-MIMO, OFDMA	2

Table 4.1: USB Wi-Fi adapters used in the emulation setup.

In addition to the physical interfaces that were selected for the emulator, it is possible to create virtual wireless interfaces from the physical WNICs, provided that the interface supports this feature. With the possibility of creating virtual wireless interfaces, it could be possible to multiply the amount of WNICs in use. With the host machine, it is possible to check whether the interface supports it or not by checking the capabilities of the physical WNIC with the iw tool available in Linux. For the chosen WNICs all of them support virtual interfaces. Only downside is that the interface might not have the capacity to act as two clients at the same time.

After testing, it was possible to create virtual interfaces of the 802.11ax and 802.11be supporting adapters. Unfortunately during the tests, the virtual WNICs did not manage to send any traffic to the network. These issues are likely to be explained by limitations in either hardware or software.

4.1.4 Host machine

When the devices are selected for the environment, some new questions will be uncovered. The way to connect many WNICs to an AP can be overwhelming. With Windows Operating System (OS) connecting multiple WNICs to a single AP can be complicated as the networking tools that can be used with Linux systems are not necessarily supported. Therefore, in this thesis, the system that acts as a host machine for the WNICs is Kali Linux 2025.3, because the Kali OS allows better possibilities for wireless configurations as the operating system is originally intended for security testing.

The host machine is used to control the WNICs providing the base for the emulator. It is possible to divide the host machine into two parts: client part and the control script part as presented in figure 4.1. The client part of the host machine includes the USB powered WNICs that are used in the emulator setup of this thesis. The control script is utilized to control the connected WNICs to generate WLAN traffic to send to the server hosted on the server machine. An indepth description of the control script is provided in section 4.2.3.

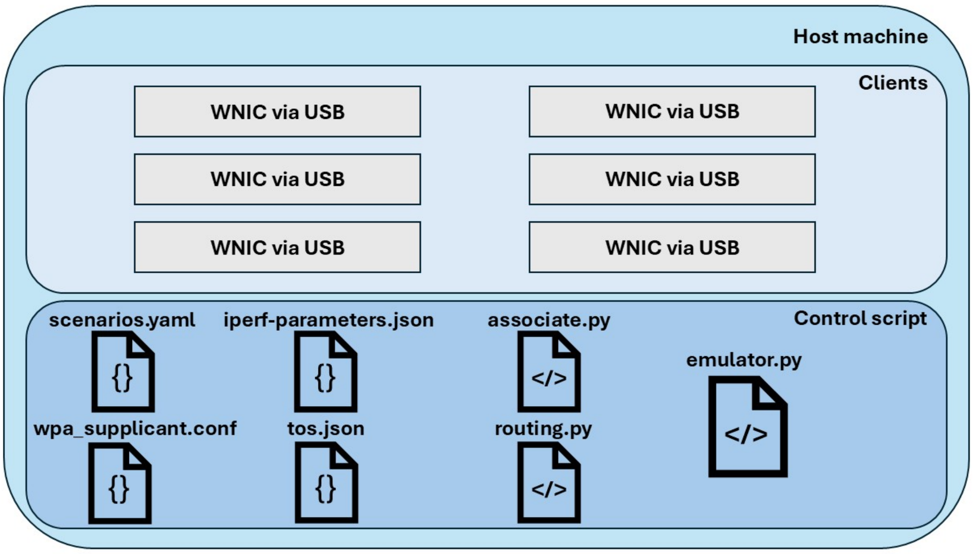

Figure 4.1: Host machine setup.

Due to low number of USB ports available on the host machine, USB hubs are utilized to connect six wireless USB powered WNICs to the host machine. These USB hubs support USB 3.1, which should provide sufficient bandwidth to support multiple WNICs. Each of the WNICs are connected to the AP to enable communication with the server machine discussed in section 4.1.1. Basically, the host machine sends packets through WNICs to an AP and the AP routes the traffic to the server machine. The whole emulation process where the host machine runs the emulator.py file and the WNICs start sending data to the iPerf2 server through the AP will be further described in section 4.3.

The emulator.py script itself is the one that reads the configuration files and executes the necessary commands. The structure and connection between the files of the control script will be discussed more in section 4.2. The laptop that is used to host all the clients and script is Dell Latitude 7340, and as discussed already it has Kali Linux as the operating system of choice. The CPU on the laptop is 13th Gen Intel Core i7-1365U with integrated graphics, which is suitable for the multi-client emulation setup of this thesis.

4.2 Software

Since the client emulator setup needs to have control over multiple WNICs, the control script needs to be well suited for all the devices. To simplify the emulation compared to the commercial products, the idea is that with the control script, it is possible to use one command to perform the necessary benchmark tests. This makes the emulation easier and more simplified compared to configuring each WNIC or device separately.

All the tools and applications used in the multi-client emulator setup can be used through terminal, and that is why the sub processes in Python are important. To make the connected interfaces on host machine act as their own clients, it is important to create the control script in a way that it handles each interface separately. The tools selected for the emulator are iPerf2 and wpa_supplicant. Before discussing the control script, it is important to understand what the iPerf2 and wpa_supplicant tools are and why they were selected.

4.2.1 iPerf

As discussed briefly in chapter 3, network traffic generation is part of the emulation setups. Therefore, it is required to select the best possible software for generating traffic. One challenge of selecting the correct software is that it should support multi-client communication, therefore multi-threading possibility is necessary for the traffic generator tool. At least, if the software has support for multi-threading, the setup becomes much simpler, since only one server is required for multiple clients.

The software chosen for the presented multi-client emulator system is called iPerf2. It has many versions, and the newest one is iPerf3, which is a refactor of the previous version iPerf2. The biggest difference between the two is that iPerf3 server accepts only one client simultaneously, while iPerf2 server can accept multiple clients simultaneously [47]. Because of this, iPerf2 was chosen for the system presented in the thesis. With multi-threading possibility iPerf2 allows the creation of multi-client scenarios, which is important for the emulation setup. Especially, when the newer IEEE 802.11 standards have features to help congestion in high density multi-client environments.

While iPerf2 is a tool for measuring maximum bandwidth in a network, it can also be suitable for emulating real world network traffic thanks to its vast number of parameters and configuration possibilities. With the parameters it is possible to control one client's bandwidth and window size. These parameters are all important when trying to create real world traffic, whether it is UDP or TCP traffic. In addition to this, iPerf2 has a parameter for ToS values, which helps to emulate for example a zoom video call where the QoS access category from table 2.2 in section 2.2 would be video. With a simple configuration file, users can set up the clients in a way that best suits their needs.

4.2.2 Associating clients through wpa_supplicant

To simplify connecting the emulated clients to the AP, some command line program is required. This way it can be simply run in the sub process of the control script. Linux systems have many options for associating clients to an AP and the option that was selected for the emulation setup is wpa_supplicant. It is part of a project that also includes the hostapd user space daemon for creating access point out of a network interface [48]. The wpa_supplicant part of the project is well suited for the emulation setup because it enables the possibility to connect clients to an AP through the command line.

With wpa_supplicant, configuration files are required to have the correct configurations for AP setup. It is possible to create multiple configuration files and that way have access to different APs. The way the configuration files are set up in this emulation setup, there are three configuration files. One is for 2.4GHz communication, second for 5GHz and third for 6GHz communication. The 2.4 and 5GHz configuration files are quite similar, the only difference being the SSID of the network, since it is possible to separate the bands on the AP used in the emulation setup. The 6GHz configuration setup requires more configuring, since the band only supports WPA3 security, which needs to be specified in the configuration of the wpa_supplicant.

The wpa_supplicant also helps in the multi-client scenarios since each client needs to be associated separately. This means that it is possible to even use different configuration files, if the user wants to see how devices act when connected to different bands. Generally, wpa_supplicant file has a lot of configuration possibilities, and especially for future development of the tool these possibilities might be suitable.

4.2.3 Control script for multi-client emulator

The control script that manages everything is written in Python, which was chosen because it is simple and it is easy to use the sub processes to run the necessary commands for the emulation. Python is efficient at reading the configuration files, which is another reason for using Python. The structure of the script can be seen in figure 4.2.

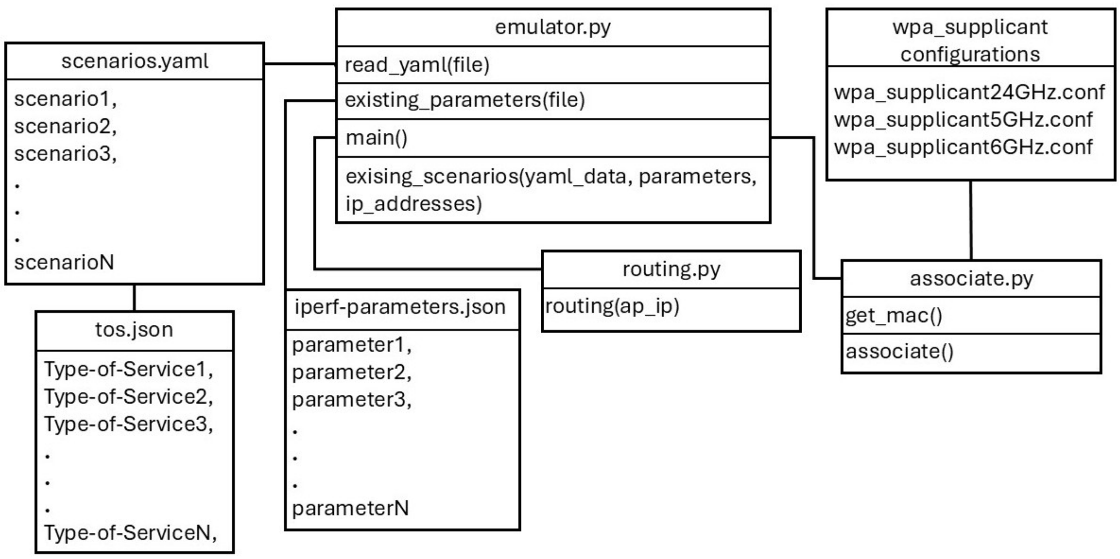

Figure 4.2: The structure of the script.

The script part of the emulation setup evolves around the emulator.py python script. As can be seen in figure 4.2, the emulator.py file contains few functions that control the host machine and the WNICs connected to it. The first function seen in the emulator.py file is called read_yaml. This function will read the scenarios.yaml file, which contains the parameters for the iPerf2 commands that will be used to create the wanted environment. The read_yaml function then returns the yaml data to be used later in the script.

The file that is given as a parameter for the read_yaml function can be seen in the figure 4.2 as scenarios.yaml. This file contains scenarios that can be used for creating the WLAN environment and communication. Scenarios.yaml file contains the parameters for the iPerf2 and all the values to use for the parameters for each WNIC connected to the host machine. This scenarios file is critical for the reliability of benchmark testing, since by utilizing this, it is possible to reproduce the same tests. Multiple scenarios can be created for the emulation setup. One example of a scenario is presented in listing 4.1.

Listing 4.1: Example of a configuration in scenario.yaml file.

```
scenario1:
    clients:
    - client:
        protocol: tcp
        mac: WNIC1 MAC
        port: 5001
        window size: 8M
        tos: 0x08
    - client:
        protocol: tcp
        mac: WNIC2 MAC
        port: 5002
        tos: 0xe0
        time: 15
```

As can be seen in listing 4.1, there are many iPerf2 parameters and values that can be given, making it possible for the user to customize the behavior of the WNICs accordingly. In addition, there is value for Type of Service (ToS), this value can be selected from a file called tos.json as can be seen in figure 4.2. With the ToS value the user can affect on the traffic since it controls the priority level of QoS for the traffic. If for some reason iPerf2 parameters for a client are left empty, the iPerf2 will use all the default configurations for connecting to the iPerf2 server.

The emulator.py file also has a function for reading the existing iPerf2 parameters. This function existing_parameters will read iperf-paramters.json file that has all the iPerf2 parameters listed from the iPerf2 documentation [47]. The function will return the JSON in a readable form since it will be used later in the script.

For the main function inside the emulator.py file, there are few additional scripts that are used to start the emulation process. First, the emulated clients need to be associated with the AP, this is done in the associate.py file. This is done separately because it enables the possibility to test only association to an AP, and with that the AP can be stress tested with multiple clients reconnecting. For the emulation purpose, the associate.py file has couple functions. The get_mac function fetches all the MAC addresses of the WNICs that are connected to the host machine. These are then utilized to tie the scenarios to certain emulated clients. The associate function is the part that establishes the connection to the AP with the use of wpa_supplicant and the created wpa_supplicant.conf files as seen in figure 4.2. Only one of the wpa_supplicant configuration files can be used for running the emulation process, and the user can select it after running the control script before the emulation process starts.

In the association function the IP addresses for the emulated clients are also assigned through DHCP server on the AP. Once the IP addresses have been assigned, the script will continue with the main function and the routing.py file will start its process. The routing function will require the IP address of the WNICs and the AP, because the emulated clients require commands to force communication through specific routes on the host machine. This issue and solution will be discussed more in section 4.4.2.

Once the connection and routes are established on the host machine, the main function will continue to call the existing_scenarios function. This function gets the scenarios.yaml and iperf-parameters.json data as parameters as well as the IP addresses of the emulated clients through the associate.py script. The existing_scenarios function compares all the data from the configuration files and selects all the correct parameters and values for the execution according to the scenarios.yaml file configurations. For each connected WNIC, the iPerf2 commands are formed and stored in a variable. Once the whole scenario of the scenarios.yaml file is gathered and the commands for all the emulated clients are ready, the script will execute the iPerf2 commands. Once the iPerf2 processes are done, the program can be stopped or executed again from the beginning with the same or another scenario.

4.3 The multi-client emulation process

While finding Linux compatible WNICs can get overwhelming, the control script discussed in 4.2.3 will provide easy control over the chosen WNICs for emulating the WLAN environment. The idea is that there are pre-created configurations that manage the required parameters for emulating the WLAN environment. The emulation setup was tested with maximum of six WNICs connected to the host machine. The whole setup and data flow can be seen in figure 4.3, where host machine has run the control script and initiates the association of interfaces to the AP and fetches IP addresses for the WNICs. There is another computer connected to the AP with Ethernet cable to act as a server machine for the clients of the host machine to connect to.

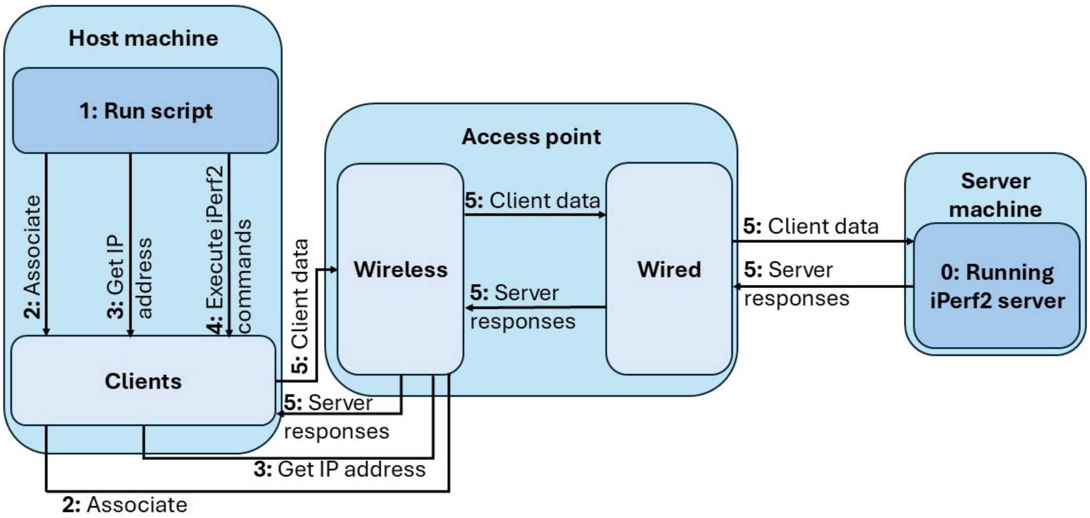

Figure 4.3: The emulation process of the multi-client emulator.

In figure 4.3, there are numbered parts to describe the flow of the data and how the emulation process generally proceeds. There are three main parts in the setup, host machine, wireless AP and server machine, which were discussed in sections 4.1 and 4.2. The emulation process proceeds as follows:

0. Before starting the emulator, it is required that there is a machine hosting a server that the host machine can communicate with through WNICs. For running the emulation process the first step will be to start the iPerf2 server on the server machine.

1. On the host machine, the created control script can be run after starting the iPerf2 server on the server machine. Running the control script will initiate the emulation process, and the script will start by controlling the actions of the connected WNICs. The WLAN environment is created in accordance with the configuration files provided in the script.

2. To start communicating with the server machine, the WNICs need to be connected to the same AP as the server machine. At this point of the emulation

WNICs are associated with the AP, as seen in the figure 4.3.

3. After WNICs are associated with the AP, they require an IP address for sending client data in the network. This will be done with the help of dhclient tool on the host machine, which fetches the unique IP address from the DHCP server of the AP for each WNIC connected to the host machine.

4. Once the WNICs are associated with the AP and IP addresses are assigned, communication with the iPerf2 server can be initiated. In this part, the script will read the configuration files and assign the parameters for the iPerf2 commands. The parameters will be assigned according to the configurations done for each WNIC in the configuration file. The iPerf2 command runs simultaneously for each WNIC, and the WNICs will communicate with the same iPerf2 server.

5. Lastly, the data traffic is initiated in the WLAN environment. Here the WNICs are sending the data through the iPerf2 process to the AP, which will route the client data of each iPerf2 process accordingly to the server. The server will respond to the client data of each WNIC, and the flow will continue until the last WNIC is done. After this the script can be stopped, or the environment can be emulated again.

The data sent during the emulation uses TCP or UDP. These differ in the sense that TCP is reliable connection-oriented transport layer protocol. While UDP is unreliable connectionless protocol that does not guarantee packet delivery. UDP is used mostly in video and streaming applications, while TCP is used when the data is required to be complete. It is possible to use both protocols at the same time with different clients or use only one protocol at a time.

When the emulation process is finished the data that was flowing through the environment can be stored on the host machine. This can be then analyzed further, the analyzation process will be discussed in more detail in chapter 5.

4.4 Development process

Development processes will always face some challenges. Some can be avoided easily, and others require more consideration and tinkering with the hardware or software. For this multi-client emulator setup, challenges faced during the development process were mostly related to the multi-client emulation function. It appears that handling multiple emulated clients can be tricky even when they are controlled with one device.

4.4.1 Multiple clients

One of the biggest challenges for creating this multi-client emulator setup is that handling multiple clients is overwhelming and complicated. While the script offers a possibility to control multiple clients on a simple interface, some issues with multiple clients still occur. Although the issues faced with the multi-client setup were not critical, they needed to be addressed in order to get the emulation setup working.

The initial plan for the setup was to create traffic from the ground up. This would have been done by creating sockets in Python and connecting the clients to a server, similar as the current setup does. This would have enabled possibility to have different types of traffic since it would have been possible to have, for example, video call between the server and client. This plan faced a lot of issues with some multi-threading problems and was therefore not a viable option for multi-client emulation. This was until the iPerf2 was found to be capable of multi-threading and multi-client communication. In addition, the Type of Service possibilities with iPerf2 tool in general enables the traffic to emulate video or any other type of traffic in the environment.

One challenge during the development was the reproducibility of the scenarios for multiple clients. The main problem here was the handling of the interfaces, especially when they are disconnected from the host machine. When reconnecting them to the host machine, they might get different names and the order of the interfaces changes. Then the parameters from the scenarios file might be assigned to wrong interface compared to the previous emulation process. This challenge was addressed by assigning the MAC addresses of the interfaces to the scenarios file. This way the parameters are always tied to same interface and capabilities it offers for the WLAN environment. This complicates the process of adding new interfaces to the scenarios.yaml file in the control script, but it is more important to have reliable benchmarking tests and reproducibility.

4.4.2 Linux host model

During the initial execution of the multi-client setup on the Linux host machine, the system forced the use of the host machine's integrated WLAN interface, even when external USB WNICs were connected to the AP. This situation is discussed in RFC 1122 [49], and more specifically the multihoming situation, where multiple interfaces are connected to End System (ES) (i.e. host). There are two models in these types of situations one of them is weak host model and other is strong host model.

By default, Linux systems are using the weak host model as stated in RFC 6419 [50]. This becomes an issue in multi-client emulation setup because the packets can be routed only through one wireless interface disregarding the IP addresses of the packets. This is not suitable for multi-client emulation because the wireless interfaces connected to the host machine need to act independently from each other. This is where the strong host model can help to clarify the distinction between the interfaces.

To achieve the strong host model setup, Linux requires a few configurations in the IP tables, routes and rules on the host machine. It is discussed in RFC 6419 that Linux kernel can support up to 252 routing tables in addition to the local and main tables [50]. In addition, RFC 6419 states that there can be arbitrary number of rules on a routing table. This way it was possible to avoid the weak host model of the host machine and make packets flow through the corresponding independent emulated client.

The way the solution works is that each connected WNIC on the host machine gets one new IP table. After the IP table is created for a WNIC the rule to force the use of certain WNIC can be added to the new table, and this way the correct WNIC sends the correct packets that have the WNICs IP address assigned. This solution is automated in the routing.py file of the script. This way the user does not have to do this every time starting the emulation process. With this the connected interfaces can act as independent clients on the host machine.

4.4.3 Collecting comparable results from client devices

Lastly, to get results from devices with different 802.11 based standard support, to compare with the multi-client emulation was one obstacle. As already mentioned in this thesis, controlling multiple devices to connect to an AP and sending traffic can be complicated. Chapter 5 covers the results and the methods used for testing the proposed system. Main idea is that iPerf2 will be used in all the tests, whether it is results from the emulator setup or the physical client devices. This way the results gathered from the emulated environment and physical clients can be compared in the same way since all results are gathered similarly.

5 Performance evaluation and validation of the multi-client emulator

To evaluate whether the wireless traffic generated by the proposed WLAN traffic emulation setup corresponds to a real-world WLAN environment, the generated traffic is compared with traffic produced by physical WLAN devices, such as laptops and mobile phones. The emulator setup was compared against six different devices, and for each test there is a different combination of devices. The physical WLAN devices are selected so that they are similar for their wireless capabilities as the interfaces used with the emulator during the test. As the goal was to create a multi-client emulator setup, the tests require at least two clients at once to compare the results.

For each test, the data collected was generated with iPerf2. Therefore, to get similar comparable results from physical WLAN devices, they also require the use of iPerf2. The tool was downloaded for each device, and during each test the devices were controlled to act similarly to the emulator. For each test, the devices and the emulated client devices with the emulator setup have the same iPerf2 parameters to see if there are differences between them. This way it is possible to validate if the emulator setup is useful when compared to actual devices used for multi-client

WLAN environment testing.

The tests consist of three main parts. The first part contains tests for a two-client environment, and it is discussed in section 5.1. The traffic sent in the tests tries to mimic real-world traffic, to see if the emulator setup is capable to send such traffic. Each two-client test setup will be discussed more in section 5.1. The test presented in 5.2 contains a multi-client scenario of six individual devices and six interfaces on the emulator setup. The third part of the testing presented in section 5.3 does not have comparison, but the emulator setup was tested with four different WNICs to test the endurance of the setup.

5.1 Two-client environment tests

Testing the two clients is divided into three parts by the capabilities of the devices and the interfaces connected to the emulator setup. First test setup contains devices that support 802.11ac standard, second test contains devices that support 802.11ax standard and the third test contains 802.11be based devices.

To prove that the proposed emulator setup is capable of many different combinations and scenarios, the AP configurations are also modified for different tests. This way the results show that it is possible to emulate variable environments with the emulation setup just by using iPerf2 as traffic generation tool. Each physical device used in comparison to the emulator will be discussed in their own sections and the configurations for AP are introduced. For each test covered in this section, one test lasted for one minute. The one-minute mark was selected because it is enough to see stability in the connection while simulating real-world communication, since physical client devices send smaller bursts of data at a time.

To get valuable results, each test was run 10 times. From each iteration, the data from iPerf2 was collected for further analysis. The analysis was done in Excel, and the result is the average of the 10 iterations that were run for the test. The results focus on the average TCP and UDP throughput of the test, which indicates if the emulator can achieve similar throughput as the actual devices.

5.1.1 IEEE 802.11ac clients

The first test discussed is based on the 802.11ac standard devices. For this the multi-client emulation setup has two Netgear AC1200 USB adapters in use. This emulator setup will then be compared to Lenovo Tab M11 and Asus TUF Gaming FX505GT laptop sending same data as the emulator. The tablet and laptop are also 802.11ac devices, which is why they were selected to compare it with this emulator setup. For comparable results, the Lenovo tablet and Asus laptop initiated the iPerf2 tool at the same time, which makes it possible to see if they affect each other on the same WLAN. Similarly to the multi-client emulator which initiates all the iPerf2 processes for the Netgear AC1200 USB adapters at the same time. In the emulator test and the physical client test the devices are connected to an AP hosting a WLAN on 2.4GHz frequency band on channel 3.

Listing 5.1: The used parameters for Netgear AC1200 USB adapters and the physical client devices.

```
2-ac-clients:
    clients:
    - client:
        protocol: udp
        mac:
        time: 60
        len: 250
        bandwidth: 1M
        tos: 0xb8
    -client:
        protocol: tcp
        mac:
        time: 60
        parallel: 4
        window size: 4M
```

With these devices and configurations, it is possible to start the test with the parameters specified in listing 5.1. While iPerf2 is a tool for testing the maximum performance of a network, it is not the goal for this thesis. Therefore, the parameters in listing 5.1 are specified for more realistic network traffic from two clients. For emulating VoIP traffic one of the Netgear AC1200 USB adapters and Asus TUF Gaming send UDP traffic to the iPerf2 server. To limit the throughput the bandwidth parameter sets the client to aim for 1 Mbit/s throughput, which is more realistic for VoIP than maximum throughput. The ToS value sets the QoS value for the packets that iPerf2 sends. With the value set in listing 5.1 under the UDP client, the QoS priority level is 7, which fits into voice category as shown in table 2.2 in section 2.2.

Lenovo Tab M11 was specified as the TCP client for comparing with the other Netgear AC1200 USB adapter used with the emulator setup. To emulate high load TCP traffic, the iPerf2 parameter parallel creates four connections to the server. With this and the window size limited to 4MB the high load traffic is emulated with the emulator. The MAC address of the WNICs in all of the listings presented in this thesis are left out for security reasons.

The average throughput for the Lenovo tablet and Asus TUF gaming laptop is presented in figure 5.1. The tests were run with the same parameters as the emulator setup used for the same test. The two devices were difficult to maintain when compared to just running one command on the multi-client emulator setup. The results, in figure 5.1, show a lot of variability with the TCP client. This might occur because the tablet runs other processes at the same time as the tests are run, which of course affects the results. For the UDP client, as the QoS value is quite high and the throughput is small, the variability is decreased. Because the traffic was sent at the same time as the TCP and UDP clients, the figure shows some correlation when TCP throughput increases and the UDP decreases.

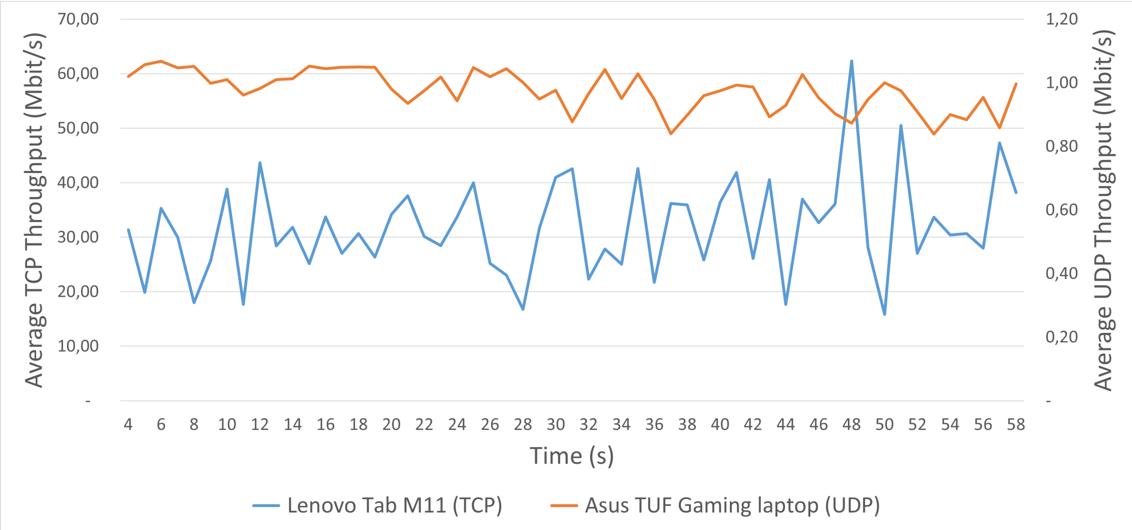

Figure 5.1: Average TCP and UDP throughput of two 802.11ac physical client devices sending iPerf2 data at the same time.

The results of the emulator can be seen in figure 5.2. The result is clearly different from the two different client devices. As can be seen in the figure 5.2, the UDP traffic is much more stable. As well as the TCP traffic of the emulated client does not have that much variability. The figure 5.2 also shows a decrease in throughput during the test. This might be because the Netgear AC1200 interfaces are not that reliable, because during the tests the Netgear AC1200 USB adapters became physically hot during the end. The reason for the stability of the result is most likely tied to the fact that the interfaces have no other processes running during the test period. Without other processes, the processing power of the Netgear AC1200 USB adapters is used completely for the emulated traffic.

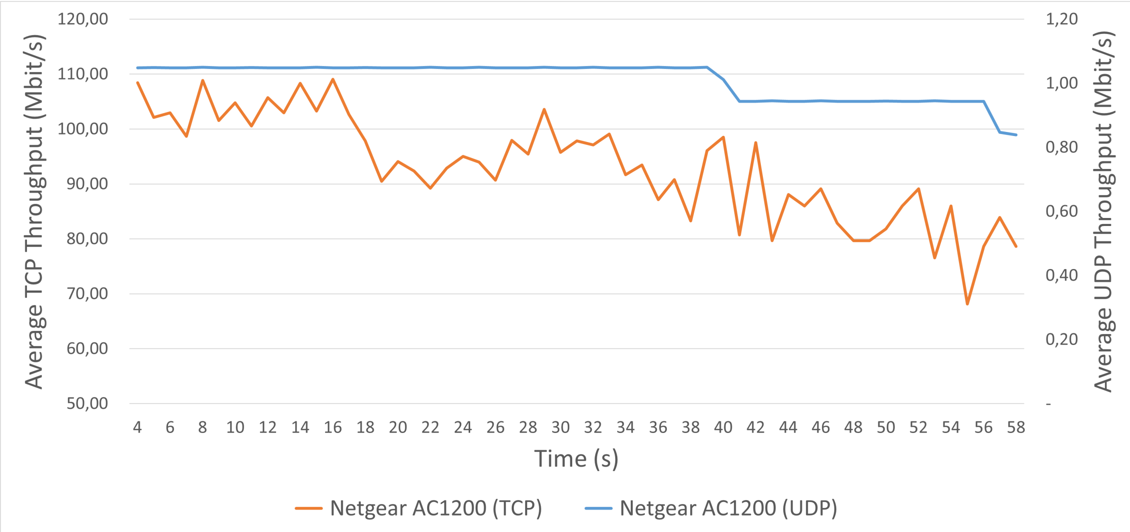

Figure 5.2: Emulated 802.11ac client devices sending TCP and UDP data at the same time.

To compare the two results, the emulator seems to generate more stable traffic to the server machine. This can be good for testing the stability of WLAN environment, since the interfaces connected to the emulator only focus on the one iPerf2 process that is running during the test. The average TCP throughput of the four parallel streams on the emulator managed to achieve higher TCP throughput which seems to indicate that it would be more powerful to use the emulator for TCP tests. The UDP clients on the other hand have a lot of differences. Where the emulator results seem to be stable, it might not be that realistic compared to the UDP traffic of the Asus laptop. Of course, the WLAN environment will affect the results, since the tests were run on different days, the environment is most certainly different. Still, from these results the emulator setup with the two Netgear AC1200 USB adapters can emulate WLAN environment of two 802.11ac client devices.

5.1.2 IEEE 802.11ax clients

Similarly to the 802.11ac based client tests, the 802.11ax client tests are run for one minute. Each test is repeated 10 times and the average TCP and UDP throughput are calculated for the results. To prove that the multi-client emulator is capable of creating variable WLAN environments, the iPerf2 parameters for this test are modified. The goal is to try to emulate gaming traffic and download traffic at the same time with two different physical clients and two Netgear Nighthawk AXE3000 USB adapters connected to the emulator setup. As discussed in section 2.5, the 802.11ax amendment improves the efficiency of wireless communication. With this the standard supports larger bandwidth and higher throughput, while granting access to the 6GHz band communication for less congested wireless traffic.

Since there is a possibility for the 6GHz band connection with the hardware that is in use, the test has two parts: tests done on 5GHz band and tests done on 6GHz band. The emulator has possibility for two WNICs in 802.11ax standard setup, and it uses the Netgear Nighthawk AXE3000 USB adapters with support for the 6GHz band. With this the results are compared to results of the physical devices which are Dell Latitude 7340 and Steam Deck OLED. The WLAN environment that is used in 5GHz tests is configured to use 5GHz frequency band on channel 40, while for test on 6GHz frequency band channel 25 is used.

With these physical client devices and emulated clients, the iPerf2 parameters were created as shown in listing 5.2. The same parameters were used for both 5GHz and 6GHz tests, this way it is possible to see if the 6GHz band affects the results. The iPerf2 parameters are similar to the parameters used in the 802.11ac tests, with adjustments made to the tests show that the emulator is capable of many different traffic types. The UDP client bandwidth is increased for the 802.11ax test and the ToS value is changed for the QoS to be high enough priority level to simulate real-time interactive traffic. The physical client device sending the UDP traffic is

Steam Deck OLED while the Dell Latitude 7340 is sending TCP traffic to the iPerf2 server. For the TCP client the iPerf2 parameters stay mostly the same with the ToS value changed to be high throughput data, and the reverse parameter changes the direction of the traffic to simulate TCP download.

Listing 5.2: The parameters used for Netgear Nighthawk AXE3000 adapters and the physical client devices.

```
2-ax-clients:
    clients:
    - client:
        protocol: udp
        mac:
        time: 60
        len: 200
        bandwidth: 2M
        tos: 0x80
    - client:
        protocol: tcp
        mac:
        parallel: 4
        time: 60
        window size: 1M
        reverse:
        tos: 0x28
```

5GHz test

The test results for the WLAN environment created on the 5GHz frequency band are presented in figure 5.3. The test results seem to be very similar to the ones gathered from the 802.11ac tests. As the average UDP throughput stays very stable the average TCP throughput has a lot of variability. The four streams emulating download traffic to the Dell Latitude 7340 laptop reaches extremely high throughput. The high TCP throughput does not affect the UDP traffic of the Steam Deck OLED in the emulated WLAN environment as can be seen in figure 5.3.

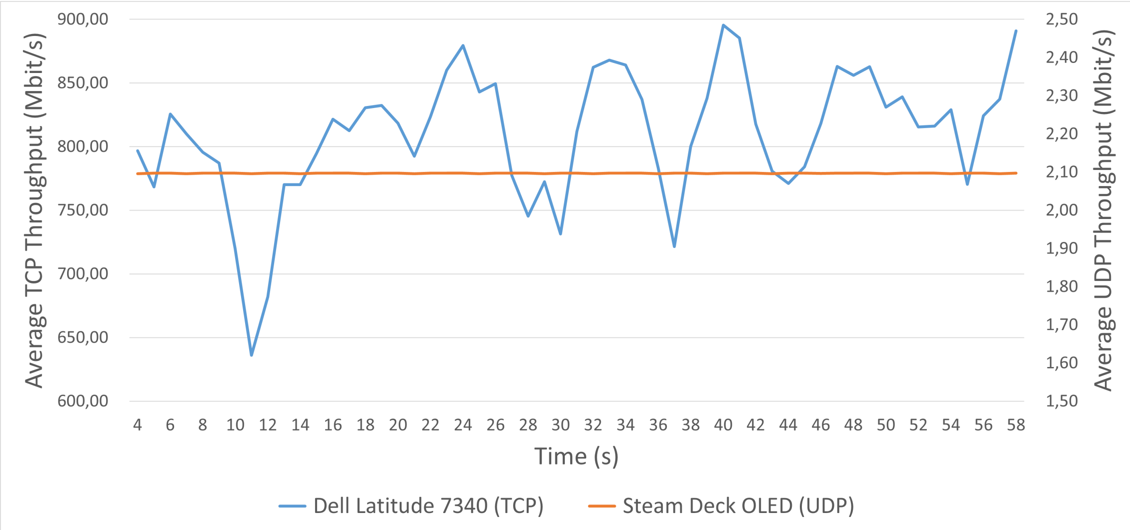

Figure 5.3: Average throughput of two 802.11ax physical client devices on 5GHz wireless network sending TCP and UDP data.

When the emulator was tested on the 5GHz wireless network, the average TCP throughput seems to be more stable compared to the TCP traffic of the Dell Latitude 7340 as seen in figure 5.4. At the same time, the average UDP throughput seems to have more variability than the Steam Deck OLED had. Still the range of the variability is so small, that the result is stable for the UDP traffic of the emulator. Also, the average TCP throughput of the emulator does not seem to get as high as the Dell Latitude 7340 average TCP throughput.

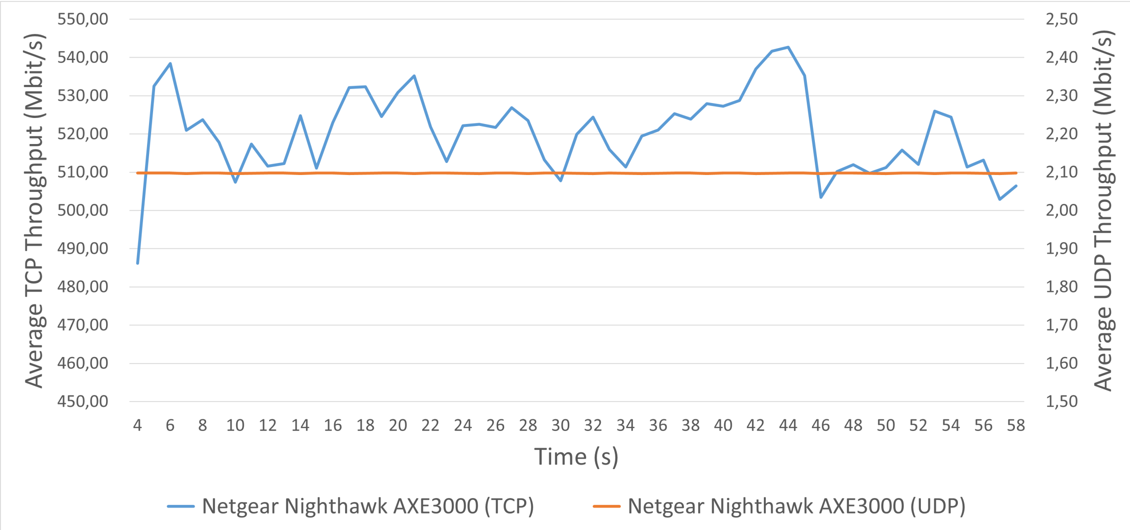

Figure 5.4: Emulated 802.11ax client devices sending TCP and UDP data at the same time on 5GHz frequency band.

To compare the results of the 5GHz wireless network tests, the emulator can act well with the 5GHz band. Even if the average TCP throughput does not achieve the same values as the Dell Latitude 7340 client, the traffic of the emulator is still valuable. The difference in the average TCP throughput might be because of the processing power that the Netgear Nighthawk AXE3000 USB adapters lack. Since the average UDP throughput in the tests is low, the USB adapter can maintain the average UDP throughput throughout the test. In the end, the emulator performs well on the 5GHz wireless network.

6GHz test

When testing on the 6GHz wireless network, the Dell Latitude 7340 seems to have decreased average TCP throughput during the test. This seems interesting, because the 6GHz band should be less congested than the 5GHz band. Still the variability of the average TCP throughput is large but seems to be bit smaller compared to the 5GHz test on the Dell Latitude 7340. The average UDP throughput seems to be similar and very stable during the 6GHz wireless network test as can be seen in figure 5.5.

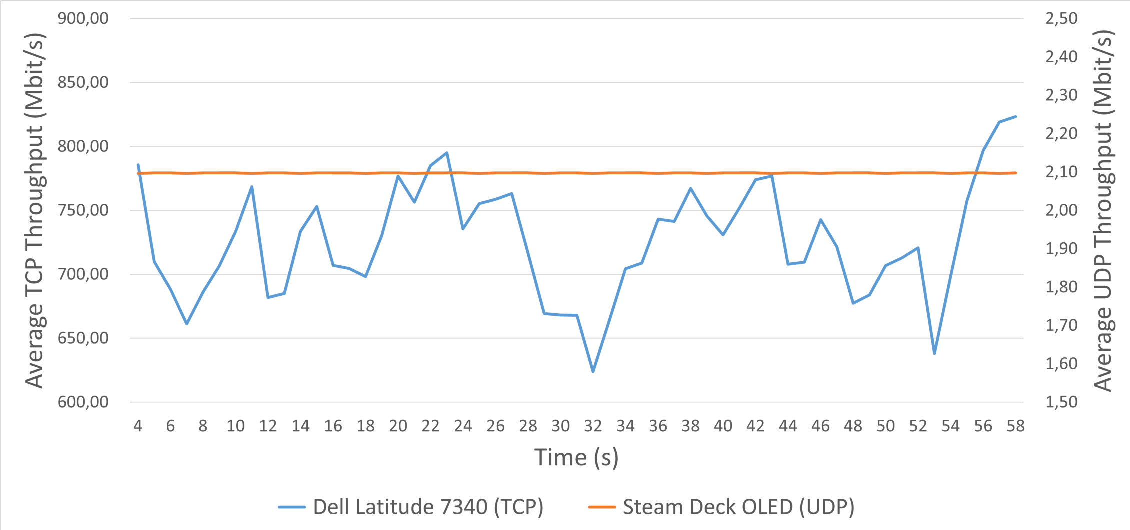

Figure 5.5: Average throughput of two 802.11ax physical client devices on 6GHz frequency band sending TCP and UDP data.

As figure 5.6 shows, the emulator has more stability in the average TCP throughput on the 6GHz wireless network. Similarly to the Dell Latitude 7340 average TCP throughput the emulator TCP throughput decreases compared to the 5GHz tests. Even when the emulator average TCP throughput on 6GHz band is compared to the 5GHz results, the 6GHz frequency band seems to offer more stability as the figure 5.6 shows.

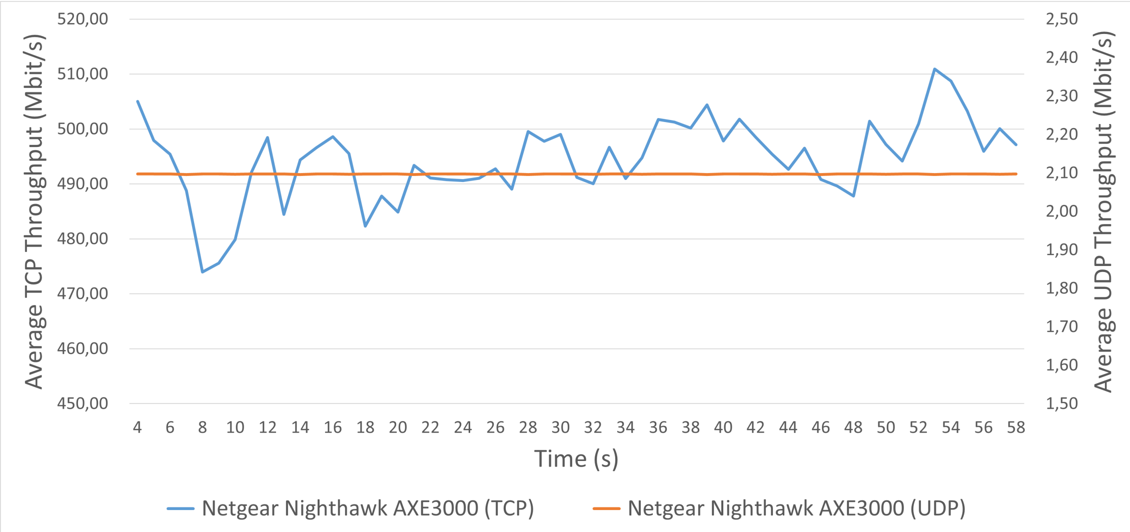

Figure 5.6: Emulated 802.11ax client devices on 6GHz wireless network sending TCP and UDP data at the same time.

The results of the 802.11ax based physical WLAN devices and the Netgear Nighthawk AXE3000 USB adapters connected to the emulator setup seem to be quite similar as seen in this section. There are some differences, which can be explained by the other processes that are running on the Dell Latitude 7340 and Steam Deck OLED at the same time as the tests are running. Still, it is possible to see that the presented emulator setup can test 802.11ax wireless environments in each of the three bands the standard supports. The test results prove that the emulator is capable of running tests with varying configurations, whether the configurations are done on the host machine or on the AP that provides the wireless network, the emulator can provide suitable benchmark test solution.

5.1.3 IEEE 802.11be clients

To test 802.11be based client devices and interfaces connected to the emulator, it is possible to enable MLO option introduced in 802.11be for the network. Since the previous tests utilized 2.4, 5 and 6GHz bands, the logical option for the 802.11be was to use MLO, which as discussed in section 2.6, can utilize multiple bands at the same time.

The configuration is done on the AP, and the MLO network needs to be enabled. The AP supports only two bands at the same time on the MLO network. The frequency bands used for the MLO network were 5GHz frequency band on channel 40 and 6GHz frequency band on channel 25. The combination of 5GHz and 6GHz band was selected for less congestion on the network. The physical client devices used to gather comparable results were Samsung Galaxy Z Flip 7 and Google Pixel 8 Pro. Both physical client devices support the 802.11be standard and the MLO network, which is why they were selected for the test. For the 802.11be client tests, the traffic for both physical client devices and for the Netgear Nighthawk BE6500 USB adapters connected to the emulator send UDP traffic to the iPerf2 server.

Listing 5.3: The used parameters for Netgear Nighthawk BE6500 adapters and the physical client devices.

```
2-be-clients:
    clients:
    - client:
        protocol: udp
        mac:
        bandwidth: 5M
        len: 1400
        time: 60
    - client:
        protocol: udp
        mac:
        bandwidth: 200k
        len: 200
        tos: 0xc0
        time: 60
```

Since the two physical client devices selected for this test are mobile phones, the traffic tries to simulate real-world mobile phone traffic with the selected iPerf2 parameters as seen in listing 5.3. Both client devices and both interfaces on the emulator send UDP data in the test result. With the limits to the bandwidth and length of the UDP the traffic tries to act as one small process that happens in real-world wireless network traffic. Again, the test is performed 10 times on each client, and the test takes one minute to run. The two phones are tested at the same time, while the emulator with two Netgear Nighthawk BE6500 USB adapters is tested separately from the two phones.

The results of the two phones running the tests at the same time can be seen in figure 5.7. As the parameters in listing 5.3 show, one of the devices has bandwidth limited to 200kbit/s. The Google Pixel 8 Pro is stable and manages to hold the 200kbit/s mark during the whole test as seen in figure 5.7. The selected bandwidth is small because it tries to emulate a situation where client device is browsing a web site for example. Same goes for the Samsung Galaxy Z Flip 7 which has a bandwidth of 5Mbit/s, which tries to emulate a scenario of video call traffic. As the range of variability of the result is small, it can be seen in figure 5.7, that the Samsung device sends stable traffic in the WLAN environment.

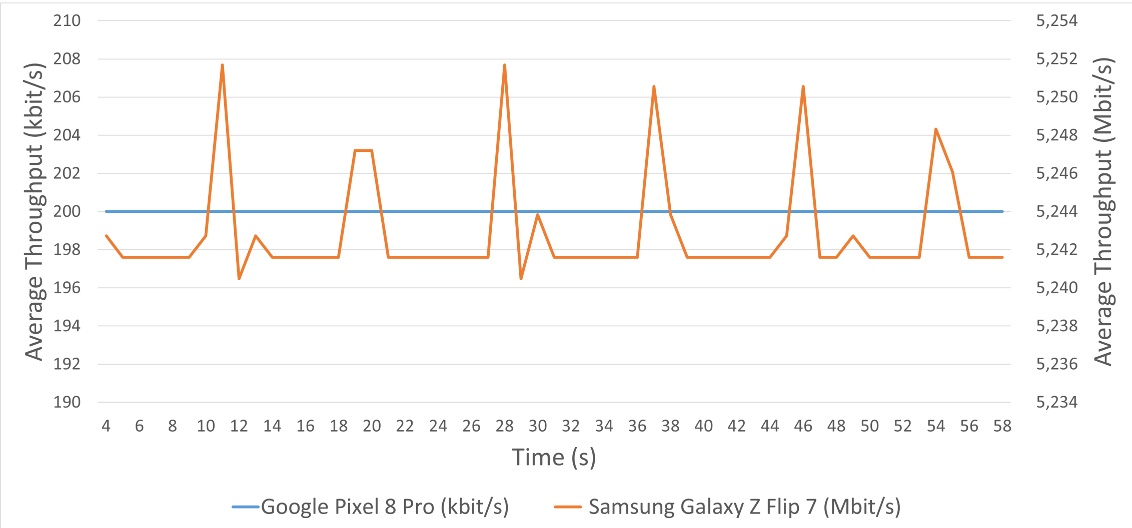

Figure 5.7: Average UDP throughput of two 802.11be physical client devices.

With the small throughput set on the iPerf2 parameters, the emulator results appear to be very similar to the client devices as can be seen in figure 5.8. The traffic seems to be stable on both the physical client devices and the emulator tests, which would indicate that the built emulator is suitable for emulating a WLAN environment with this type of traffic.

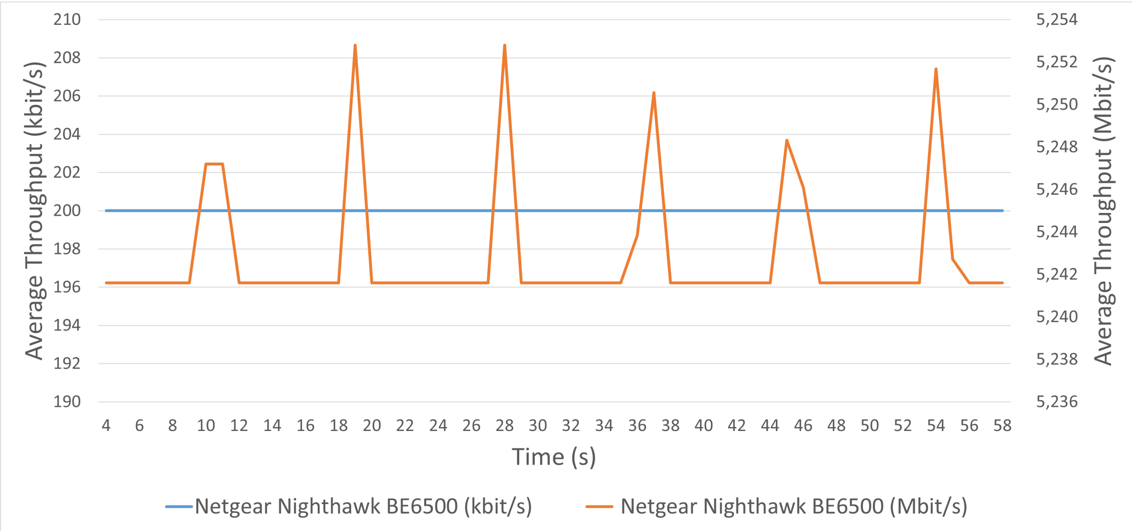

Figure 5.8: Average UDP throughput of two emulated 802.11be clients.

The done tests and analysis prove that it is possible to send TCP and UDP data with the emulator setup developed in this thesis. With the possibility of modifying the throughput and other parameters with the iPerf2 tool, the traffic can be modified to simulate mobile phone, gaming, downloading or VoIP traffic. This enables the possibility to benchmark wireless network environments, for many possible scenarios the network might face. With the possibility of emulating multiple clients, the possibilities are extensive with the created emulator setup.

5.2 Multi-client environment test

The second part of the testing process covers whether the emulator can emulate more than two clients at the same time. For this test, all the six wireless USB network adapters listed in table 4.1 were utilized. The results of the emulator were compared to new results gathered from the six physical client devices that were also used in the tests of section 5.1. On all the six physical client devices the iPerf2 tool was run at the same time to see if the devices might affect each other during the test. The emulator of course processed all the connected WNICs at the same time, thanks to the control script and setup created in this thesis.

The test was run 10 times, and the average throughput for each client was calculated for getting the most reliable result. There were no limits to the bandwidth with the iPerf2 parameters, and the iPerf2 was only set so that the test ran simultaneously for one minute. The physical client devices and the WNICs connected to the emulator were connected to a 5GHz frequency band. The 5GHz frequency band was selected because it has less congestion than 2.4GHz frequency band, and the 6GHz frequency band cannot be utilized since not all the client devices and wireless USB adapters support the 6GHz band.

The test was complicated to run on the six physical client devices when each client required attention at the same time. The result gathered of the average throughput of all the six physical client devices can be seen in figure 5.9. The traffic sent is TCP because that is the default for iPerf2. As can be seen from the figures, the Lenovo Tab M11 has higher throughput than the other clients. There can be many different factors for why that is, but the focus for the test is how stable the average throughput is for the six different clients. There are few points where, for example, Dell Latitude 7340 laptop has stability issues, but in the end all the client traffic seems to be stable.

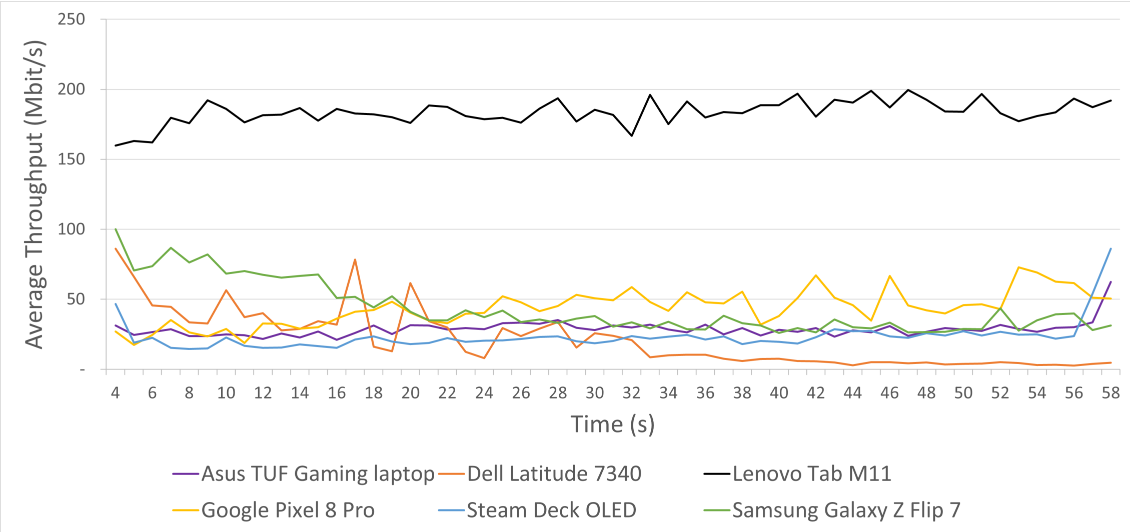

Figure 5.9: Average throughput of six different physical client devices sending TCP data at the same time.

The emulator managed to run test for each emulated client automatically by running the control script presented in section 4.2. From the results in figure 5.10, it is possible to see that Netgear AC1200 USB adapters connected to the emulator seem to achieve low average throughput throughout the tests. While the other interfaces achieve higher average throughput, they seem to have more variability. Still, when emulating six clients with the emulator, the process works well and stable enough to be used for experimenting with WLAN environment.

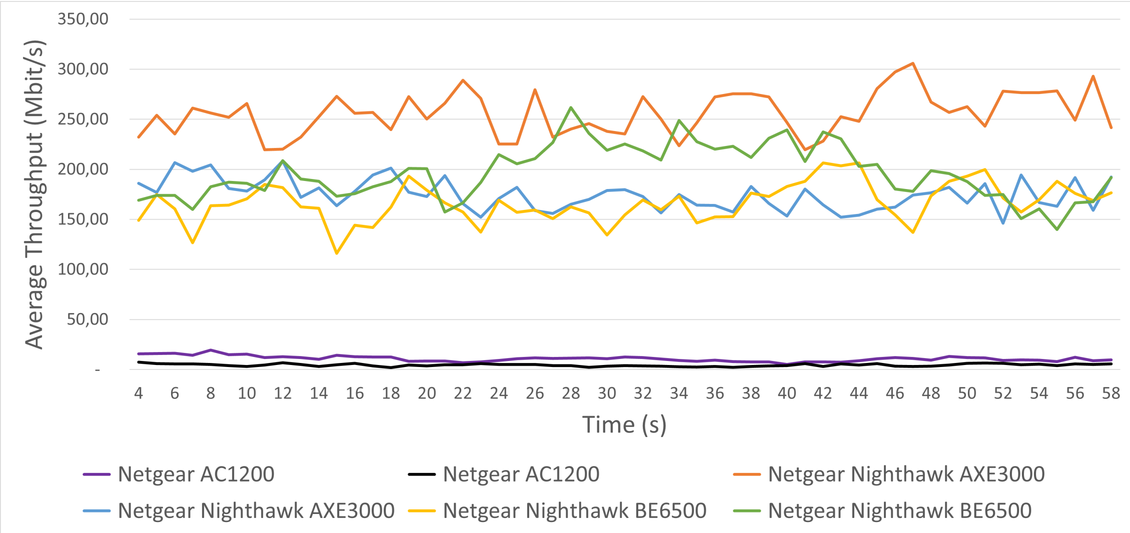

Figure 5.10: Average throughput of six emulated clients sending TCP data at the same time.

To compare the two results, the stability of the average throughput between the clients is better on the test run with the six different physical client devices. The average throughput of the emulator test seems to be higher, which could be beneficial to see for example how an AP manages the load of multiple clients. To test the emulator with more interfaces was not possible during the process of this thesis, but it truly seems that the emulator setup created could handle more clients than six. This would be beneficial for benchmarking the wireless networks in a classroom for example, or conference rooms with multiple wireless devices communicating at the same time. The results from the test run with six clients seem to confirm that the created emulator setup is in fact capable of emulating multi-client scenarios in WLAN environment.

5.3 Testing the endurance of the emulator

Wireless environment changes constantly, new devices are connected and a lot of data is propagating through the air from device to device. Because of the constant change of the wireless environment, it is important to be able to test long periods of devices sending data. For example, one interesting aspect would be to see how an AP acts when many devices send high throughput data constantly for a long period. In addition, it is possible to analyze how, for example, laptops or mobile phones react to the environment when there is constant high throughput data for long period of time. For this kind of scenario it is important that the multi-client emulator can endure situations where multiple clients send data for long period of time.

To test the endurance of the multi-client emulator, four clients were used for emulating high throughput data transfer for long period of time. The iPerf2 parameters were again defaults and the emulated clients were connected to 5GHz network which was running on channel 40. The time for the test is five hours, in that time the emulated clients were assigned to gather data for every five minutes. The interfaces used for the test were both Netgear Nighthawk AXE3000 USB adapters and both Netgear Nighthawk BE6500 USB adapters. These were selected because they support 802.11ax and 802.11be standards which handle the high throughput data better and the standards support higher throughput as discussed in sections 2.5 and 2.6.

The results of the test can be seen in figure 5.11. The test took five hours, and the results seemed to stay stable for the whole time the emulator was running. This indicates that the multi-client emulator can perform tests overnight or at least for many hours. This is something that the commercial products also offer, and therefore it is important to note that the multi-client emulator created in this thesis can perform similar tests. One important notice during the test was that the wireless

USB adapters were physically warm after the five-hour test. This might limit the use of this setup for even longer test periods, but when the interfaces used are improved the possibility of even longer tests increases with the multi-client emulator setup.

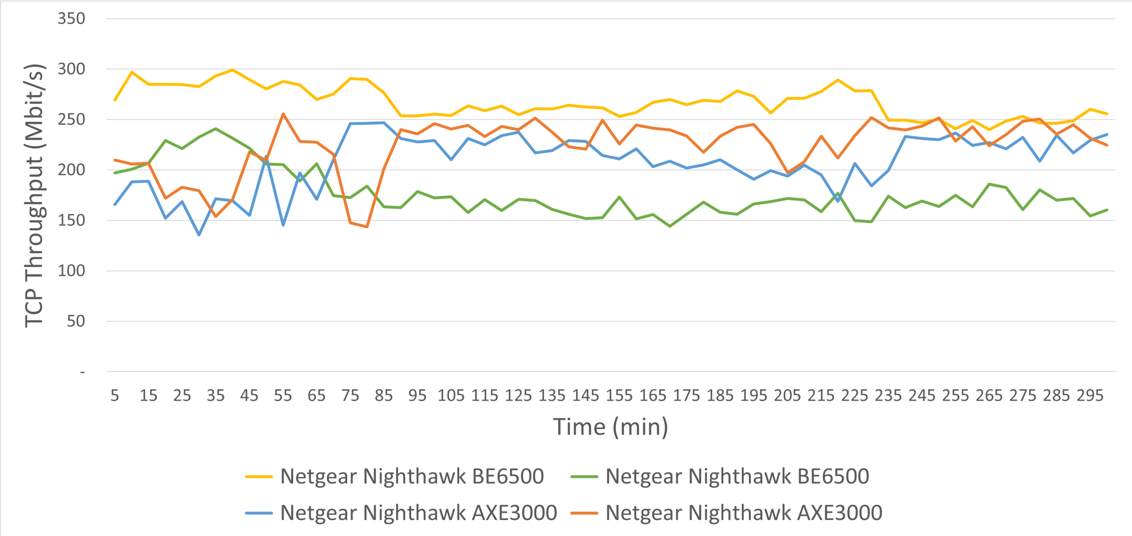

Figure 5.11: Endurance of the emulator while emulating four TCP clients for five hours.

5.4 Discussion

The tests run for the validation of the multi-client emulator setup try to cover multiple WLAN environments that can be created with the setup. The results show that it is possible to have connectivity with the newest 802.11 standards, which is important when considering the discussion of advancements made in the 802.11 standard family in chapter 2. The 802.11 standard family evolves constantly, and each standard has their own capabilities to offer, therefore multi-client emulation needs to be able to specify certain 802.11 standards to be used. Not only create WLAN environment with one 802.11 standard, but the multi-client emulator creates possibility to use multiple interfaces with different standards at the same time. This is something that commercial products most likely can produce as well, and it is important for validating how an AP acts under different scenarios.

When comparing the results of sections 5.1, 5.2 and 5.3, it seems that the proposed multi-client emulator can be used to emulate variable WLAN environments and data rates. The emulator can even offer more stable traffic compared to physical client devices since they only have the iPerf2 process running that is defined by the control script and configuration files. The emulator setup was also much easier to use than controlling the two, six or four physical client devices at the same time. On the emulator, only one command is required, and the setup runs the tests automatically. This speeds up the process and enables the user to focus on something else, while the emulation is running.

Not only do the results show that the multi-client emulator can utilize multiple 802.11 standards, but it compares well with real-world wireless traffic generated with physical WLAN client devices that were used in the comparison. As one of the main goals when creating the multi-client emulator, the emulated environment needs to be as similar as with real-world devices. Therefore, the results gathered in the tests seem to prove that this is possible with the multi-client emulator. There were some differences in the average throughput in the tests, but generally it seems that the multi-client emulator can emulate real-world scenarios well.

The multi-client emulator can also emulate different WLAN environments in the sense that emulated clients can send UDP and TCP traffic. With iPerf2 this is made possible especially when creating high density wireless scenarios. The parameters that iPerf2 offers are well suited for the multi-client emulator, where users want to try to emulate WLAN environment where emulated clients send VoIP traffic for example. In addition, the iPerf2 parameters can manage the QoS level of the emulated client traffic with the emulator. As the results prove all the parameters and interfaces can be mixed and depending on the configuration file, the WLAN environments to be created have a lot of options for many different WLAN environments.

When comparing the results of the proposed multi-client emulator with the emulators proposed in academic research discussed in section 3.1, the conclusions are similar. For example, the results in the emulator proposed by Alvarez et al. in [25] achieve similar results when compared to real-world devices. This conclusion by Alvarez et al. amplifies the results presented in the multi-client emulator proposed in this thesis. The iPerf2 tool used in the proposed multi-client emulator is proven to be useful tool in emulating real-world WLAN environments. In all of the academic research discussed in section 3.1, the iPerf2 or iPerf3 tool was utilized somehow. The deployability of the proposed multi-client emulator is also achieved, similar to the proposed solutions in [25]-[27]. The limit of emulated clients with the proposed multi-client emulator was six in this thesis. As mentioned in section 5.2 the proposed multi-client emulator seems to be able to support more WNICs if required. This statement can be verified by the academic research discussed in section 3.1, because for example in proposed emulator by Capdehourat et al. [26] they achieved similar results by emulating 20 clients when compared to the six emulated clients in this thesis.

It is difficult to compare the multi-client emulator to the commercial products since the commercial products were not available to use. For the results it would have been interesting to compare the performance of the presented system to the commercially available solutions. The hardware and software of the commercial products is more advanced and therefore they can offer more scenarios and emulations for testing. For example, the multi-client emulator in this thesis cannot emulate 500 clients as the WaveTest 6 discussed in section 3.2.2. As some commercial products were also discussed to offer VoWiFi calling and possibility of the 320MHz with the 802.11be standard, the multi-client emulator setup cannot manage these. For the 320MHz channel width, the wireless USB adapters could not utilize the whole channel, but in the future with newer and more advanced interfaces this could be possible with the emulator setup created in this thesis.

While the commercial products can offer the possibility to emulate hundreds or even thousands of clients, the multi-client emulator setup is not yet capable of such high volume of clients. While the goal was to create more affordable emulator setup, it might not even be possible without investing more in capable hardware. The control script created for the multi-client emulator should be able to manage many interfaces, which could enable the possibility for hundreds of emulated clients. As the results show the multi-client emulator can create many scenarios for long periods of time, the end goal was achieved with more affordable setup.

6 Conclusion

With the continuous development of IEEE 802.11 WLAN standard, the WLAN environment becomes more stable and capable of transferring high throughput traffic with high data rate. This is already seen with the latest 802.11be standard, which delivers significant improvements for WLAN communication between client and AP devices. New technologies are introduced constantly with the development of the 802.11 standard, such as MLO, which allows the simultaneous use of multi-band communication for client devices. The development of new 802.11bn standard has already begun, and with that the aim is to provide Ultra-High Reliability (UHR) for WLAN communication.

The purpose of this thesis was to create a emulator for emulating multi-client WLAN environments to test latest technologies and 802.11 WLAN standards. One of the main challenges when starting the planning was to gather the necessary hardware for the emulator. Eventually, total of six USB powered WNICs were used for the tests. Tests run for the thesis are considered so that the results show as much versatility as possible from the multi-client emulator. As tests presented in chapter 5 demonstrate, it is possible to emulate many WLAN environments with the created multi-client emulator. The multi-client emulator can handle all possible frequency bands and the technologies they require to connect to networks on different frequencies. Some issues occurred during the development process, for example, regarding the Linux host model. The issues did not affect the end result, and the development of the multi-client emulator did not have any setbacks.

In this thesis a multi-client emulator tool is introduced with the purpose of offering suitable tool for emulating WLAN environment to test the latest technologies and 802.11 WLAN standards. The multi-client emulator tool offers a feasible option for more advanced commercial products offered by organizations in the wireless communication field. One of the main goals for the multi-client emulator tool was to offer flexibility and stability for the user. This goal was achieved, and the user has the possibility to decide what computer to use as the host machine and what hardware to utilize with the host machine for the multi-client emulator tool. This achievement with the software part of the tool is extremely important, especially when new 802.11 standards are introduced.

The multi-client emulator tool itself requires a simple design that is sustainable and suitable for the WLAN environment under test. There were many things to consider on the hardware side such as what 802.11 based standard should the WNICs connected to the host machine support. It is important that the WNICs are suitable for the specific WLAN environment under test. As pointed out in chapter 5, the multi-client emulator tool can use WNICs that support 802.11 standards.

In order to fulfill the objective of proposing multi-client emulator for WLAN environments in this thesis, three research questions were formed:

1. What hardware is required to simplify WLAN networking emulation without having to gather multiple physical client devices?

2. How to connect multiple WNICs to a single AP for emulating a multi-client WLAN environment?

3. What kind of traffic can be sent with the emulated multi-client WLAN environment?

Research question 1 was discussed in chapter 3, where the discussion focused on the related academic research and existing commercial products. Based on these existing solutions the research question one can be answered. The WLAN networking emulation does not require expensive hardware in order to emulate a WLAN environment. The hardware required would be a computer and multiple WNICs. With this hardware, it is possible to use the selected computer to control all the connected interfaces and make them act as multiple clients connecting to an AP. By creating emulation setup based on a single computer and multiple WNICs, the emulation process is considerably simplified, as the need to manage multiple physical client devices separately is eliminated.

Chapter 4 introduces the created multi-client emulator and how it connects the WNICs to a single AP. Based on the chapter, the research question 2 can be answered. Many tools are required and the script created for the multi-client emulator setup in this thesis is useful, for connecting WNICs to an AP. The requirements for connecting the WNICs to an AP are a host machine, for example, a laptop that has the WNICs connected. Once the WNICs are connected to the host machine, it is possible to connect them to the AP by using the Linux wpa_supplicant. It is also possible to connect the WNICs to the AP by using the operating systems UI, but that takes away the simplicity of the created multi-client emulator tool that can automate association for multiple WNICs at the same time to an AP.

With the created multi-client emulator, it is possible to create many variable WLAN environments as discussed in chapter 5. To answer to research question 3 based on the chapter it can be summarized that it is possible to send TCP and UDP data with the emulator setup created in this thesis. With the possibility of modifying the throughput and other parameters with the iPerf2 tool, the traffic can be modified to emulate mobile phone, gaming, downloading or VoIP traffic. This enables the possibility to benchmark WLAN environments, for many possible scenarios the network might face. With the possibility of emulating multiple clients, the possibilities are extensive with the created emulator setup. With the iPerf2 tool integrated into the multi-client emulation tool, the possibilities to configure bandwidth and window size are introduced. This can be beneficial especially if the WLAN environment under test is specific to certain user needs, for example, classroom with many client devices sending data.

While academic research has proposed solutions for WLAN environment testing with multi-client emulator setup, the tool proposed in this thesis utilizes the latest 802.11 standards. As it stands, the multi-client emulator tool introduced in this thesis is an improvement compared to the proposed multi-client emulator tools in academic research in recent history. Not only is the proposed multi-client emulator tool improvement to academic research, but it can also be a suitable option for testing WLAN environment compared to the commercial products available. The proposed multi-client emulator has similar features compared to the commercial products, and the proposed tool has room for improvements with updates on the hardware side. This makes the proposed multi-client emulator more configurable and flexible while keeping the cost of WLAN environment benchmarking low.

6.1 Future development

There is always room for improvement and that is the case with the proposed multi-client emulator. There are improvements that can be made to both hardware and software to make the multi-client emulator more capable and easier to use. The hardware side improvements are dependent on the host machine, and how capable it is in hosting the WNICs for emulating multiple clients in a WLAN environment. By adding more USB ports, it is possible to increase the number of clients for the multi-client emulator. In addition, if the host machine supports the PCIe devices, there are splitters that can multiple the connections. With this it would be possible to utilize many WNICs, and possibly even use more capable antennas for emulating the clients in WLAN environment.

For the software side of the improvements, the multi-client emulator tool's script is not limited to the setup proposed in this thesis. With the introduction of a user interface design and implementation, the tool would be easier to use for handling the configuration files in the script. In addition to making the usability of the multi-client emulator tool easier, there is room for improvement in managing the data collected from the multi-client emulator tool. Like the commercial products that can generate graphs automatically from the data, the proposed multi-client emulator can be improved to automate the collection of the data. For more precise packet analysis, integration with Wireshark's packet capturing possibility would be useful, for example, seeing how the RU allocation is done in 802.11be WLAN environment. These improvements would enhance the benchmarking and make the analysis faster for the WLAN environment under test.

For now, the proposed multi-client emulator setup for benchmarking different WLAN environments is a suitable solution with the possibility to customize and improve the performance for specific cases. The results show that the multi-client emulator can emulate many WLAN scenarios with room to improve.

References

[1] Wi-Fi Alliance, "Global economic value of wi-fi_2021-2025", *Sep 2021*, Accessed: Oct. 29, 2025. [Online]. Available: https://6ghz.info/wp-content/uploads/2022/02/Global_Economic_Value_of_Wi-Fi_2021-2025_202109-1.pdf.

[2] S. U. Rehman, T. Turletti, and W. Dabbous, "Benchmarking in wireless networks", report, Oct. 28, 2010, p. 16. Accessed: Oct. 2, 2025. [Online]. Available: https://inria.hal.science/inria-00530329.

[3] D. Law, D. Dove, J. D'Ambrosia, M. Hajduczenia, M. Laubach, and S. Carlson, "Evolution of ethernet standards in the IEEE 802.3 working group", *IEEE Communications Magazine*, vol. 51, no. 8, pp. 88–96, Aug. 2013, ISSN: 1558-1896. DOI: 10.1109/MCOM.2013.6576344.

[4] P. C. Jain, "Recent trends in next generation terabit ethernet and gigabit wireless local area network", in *2016 International Conference on Signal Processing and Communication (ICSC)*, Noida, India, Dec. 2016, pp. 106–110. DOI: 10.1109/ICSPCom.2016.7980557.

[5] G. Geraci, F. Meneghello, F. Wilhelmi, D. Lopez-Perez, I. Val, L. G. Giordano, C. Cordeiro, M. Ghosh, E. Knightly, and B. Bellalta, *Wi-fi: Twenty-five years and counting*, Jul. 13, 2025. DOI: 10.48550/arXiv.2507.09613. arXiv: 2507.09613[cs].

[6] S. Banerji and R. S. Chowdhury, "On IEEE 802.11: Wireless LAN technology", *International Journal of Mobile Network Communications & Telematics*, vol. 3, no. 4, pp. 45–64, Aug. 31, 2013, ISSN: 18395678. DOI: 10.5121/ijmnct.2013.3405.

[7] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe", *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, Jan. 2010, ISSN: 1558-1896. DOI: 10.1109/MCOM.2010.5394032.

[8] N. Elhilali, M. Badri, and M. F. Bouami, "Evaluation of QoS over IEEE 802.11 wireless network in the implementation of internet protocols mobility supporting", *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, 2023, ISSN: 21565570, 2158107X. DOI: 10.14569/IJACSA.2023.0140237.

[9] M. A. Al-Maqri, M. A. Alrshah, and M. Othman, "Review on QoS provisioning approaches for supporting video traffic in IEEE802.11e: Challenges and issues", *IEEE Access*, vol. 6, pp. 55 202–55 219, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2872770.

[10] E. Perahia, "IEEE 802.11n development: History, process, and technology", *IEEE Communications Magazine*, vol. 46, no. 7, pp. 48–55, Jul. 2008, ISSN: 1558-1896. DOI: 10.1109/MCOM.2008.4557042.

[11] T. Paul and T. Ogunfunmi, "Wireless LAN comes of age: Understanding the IEEE 802.11n amendment", *IEEE Circuits and Systems Magazine*, vol. 8, no. 1, pp. 28–54, 2008, ISSN: 1558-0830. DOI: 10.1109/MCAS.2008.915504.

[12] IEEE, "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC)and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput", *IEEE Std 802.11n-2009 (Amendment*

to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), pp. 1–565, Oct. 2009. DOI: 10.1109/IEEESTD.2009.5307322.

[13] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "The impact of channel bonding on 802.11n network management", in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, ser. CoNEXT '11, Tokyo, Japan, 2011, pp. 1–12, ISBN: 978-1-4503-1041-3. DOI: 10.1145/2079296.2079307.

[14] E. Perahia and M. X. Gong, "Gigabit wireless LANs: An overview of IEEE 802.11ac and 802.11ad", *Association for Computing Machinery SIGMOBILE Mobile Computing and Communications Review*, vol. 15, no. 3, pp. 23–33, Mar. 29, 2011, ISSN: 1559-1662. DOI: 10.1145/2073290.2073294.

[15] IEEE, "IEEE standard for information technology–telecommunications and information exchange between systems—local and metropolitan area networks–specific requirements–part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications–amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz.", *IEEE Std 802.11ac(TM)-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)*, pp. 1–425, Dec. 2013. DOI: 10.1109/IEEESTD.2013.7797535.

[16] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtilä, "IEEE 802.11ac: Enhancements for very high throughput WLANs", in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, Toronto, Canada, Sep. 2011, pp. 849–853. DOI: 10.1109/PIMRC.2011.6140087.

[17] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, 2019, ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2871099.

[18] IEEE, "IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 1: Enhancements for high-efficiency WLAN", *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, pp. 1–767, May 2021. DOI: 10.1109/IEEESTD.2021.9442429.

[19] IEEE, "IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 2: Enhancements for extremely high throughput (EHT)", *IEEE Std 802.11be-2024 (Amendment to IEEE Std 802.11-2024, as amended by 802.11bh-2024)*, pp. 1–1020, Jul. 2025. DOI: 10.1109/IEEESTD.2024.11090080.

[20] S. Chauhan, A. Sharma, S. Pandey, K. N. Rao, and P. Kumar, "IEEE 802.11be: A review on wi-fi 7 use cases", in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, Sep. 2021, pp. 1–7. DOI: 10.1109/ICRITO51393.2021.9596344.

[21] C. Deng, X. Fang, X. Han, X. Wang, L. Yan, R. He, Y. Long, and Y. Guo, "IEEE 802.11be wi-fi 7: New challenges and opportunities", *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2136–2166, 2020, ISSN: 1553-877X. DOI: 10.1109/COMST.2020.3012715.

[22] C. Chen, X. Chen, D. Das, D. Akhmetov, and C. Cordeiro, "Overview and performance evaluation of wi-fi 7", *IEEE Communications Standards Magazine*, vol. 6, no. 2, pp. 12–18, Jun. 2022, ISSN: 2471-2833. DOI: 10.1109/MCOMSTD.0001.2100082.

[23] L. Galati-Giordano, G. Geraci, M. Carrascosa, and B. Bellalta, "What will wi-fi 8 be? a primer on IEEE 802.11bn ultra high reliability", *IEEE Communications Magazine*, vol. 62, no. 8, pp. 126–132, Aug. 2024, ISSN: 1558-1896. DOI: 10.1109/MCOM.001.2300728.

[24] L. Dauksa. "Use network emulation to create real-world conditions | keysight blogs", Accessed: Oct. 14, 2025. [Online]. Available: https://www.keysight.com/blogs/en/tech/educ/2024/network-emulation.

[25] G. Álvarez, M. Álvarez, F. Bagalciague, G. Capdehourat, J. González, J. Marfia, P. Porteiro, and A. Rodríguez, "Multi-client emulation platform for performance testing of high density 802.11 networks", in *2016 Wireless Days (WD)*, Toulouse, France, Mar. 2016, pp. 1–6. DOI: 10.1109/WD.2016.7461451.

[26] G. Capdehourat, G. Álvarez, M. Álvarez, P. Porteiro, and F. Bagalciague, "High density emulation platform for wi-fi performance testing", *Ad Hoc Networks*, vol. 70, pp. 1–13, Mar. 2018, ISSN: 15708705. DOI: 10.1016/j.adhoc.2017.11.007.

[27] T. Kawai, S. Kaneda, M. Takai, and H. Mineno, "A virtual WLAN device model for high-fidelity wireless network emulation", *Association for Computing Machinery Transactions on Modeling and Computer Simulations*, vol. 27, no. 3, 17:1–17:24, 2017, ISSN: 1049-3301. DOI: 10.1145/3067664.

[28] R. Liu and N. Choi, "A first look at wi-fi 6 in action: Throughput, latency, energy efficiency, and security", *Proceedings of the Association for Computing*

Machinery on Measurement and Analysis of Computing Systems, vol. 7, no. 1, 25:1–25:25, 2023. DOI: 10.1145/3579451.

[29] V. Planchart, L. Martinez, H. Ali, M. Toole, and M. Rahaim, "A low-cost open-source testbed for experimental analysis of ultra-dense wireless networks", in *Proceedings of the Int'l Association for Computing Machinery Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '23, Montreal, Quebec, Canada, 2023, pp. 93–97, ISBN: 979-8-4007-0366-9. DOI: 10.1145/3616388.3617547.

[30] L. Bonati, P. Johari, M. Polese, S. D'Oro, S. Mohanti, M. Tehrani-Moayyed, D. Villa, S. Shrivastava, C. Tassie, K. Yoder, A. Bagga, P. Patel, V. Petkov, M. Seltser, F. Restuccia, A. Gosain, K. R. Chowdhury, S. Basagni, and T. Melodia, "Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation", in *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, ISSN: 2334-3125, Los Angeles, CA, USA, Dec. 2021, pp. 105–113. DOI: 10.1109/DySPAN53946.2021.9677430.

[31] H. Mäntylä, "Wi-fi 7 performance evaluation in private wireless networks", Nov. 21, 2025. Accessed: Jan. 26, 2026. [Online]. Available: https://aaltodoc.aalto.fi/handle/123456789/141112.

[32] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, T. Alanko, and K. Raatikainen. "Seawind: A wireless network emulator", ResearchGate, Accessed: Jan. 26, 2026. [Online]. Available: https://www.researchgate.net/publication/2889162_Seawind_a_Wireless_Network_Emulator.

[33] Keysight. "How to test wi-fi 7 signaling, RF, and throughput", Keysight, Accessed: Oct. 14, 2025. [Online]. Available: https://www.keysight.com/us/en/use-cases/test-wi-fi-7-signaling-rf-and-throughput.html.

[34] Keysight. "Keysight WaveTest 6 wi-fi 6 test solution", Keysight. Section: Article Section, Accessed: Jan. 27, 2026. [Online]. Available: https://www.keysight.com/us/en/assets/3121-1106/data-sheets/Keysight-WaveTest-6-Wi-Fi-6-Test-Platform.pdf.

[35] Spirent Communications, "Spirent TestCenter", Apr. 2020. Accessed: Jan. 26, 2026. [Online]. Available: https://assets.ctfassets.net/wcxs9ap8i19s/0cM2kLkBu0DUQe4lGygQk9/d3cf97c6ed5d48ac4e8478775bbff0b8/Spirent_TestCenter_WLAN_multiclient_emulation_and_testing_datasheet.pdf.

[36] Keysight. "Wireless network emulators", Keysight. Section: Article Section, Accessed: Jan. 27, 2026. [Online]. Available: https://www.keysight.com/us/en/products/wireless-network-emulators.html.

[37] Keysight. "WaveTest 6e", Keysight. Section: Article Section, Accessed: Oct. 21, 2025. [Online]. Available: https://www.keysight.com/us/en/assets/3122-2127/data-sheets/WaveTest-6E.pdf.

[38] Keysight. "E7515w UXM wireless connectivity test platform", Keysight. Section: Article Section, Accessed: Oct. 21, 2025. [Online]. Available: https://www.keysight.com/us/en/product/E7515W/uxm-wireless-connectivity-test-platform.html.

[39] Alethea. "WiCheck: Universal wi-fi test solution | alethea communications", Alethea Communications Technologies, Accessed: Oct. 13, 2025. [Online]. Available: https://aletheatech.com/wicheck/.

[40] Alethea. "WiCheck: Wi-fi 7 functionality testing: Unlocking next-gen wireless performance", Alethea Communications Technologies, Accessed: Feb. 2, 2026. [Online]. Available: https://aletheatech.com/wicheck-wi-fi-7-functionality-testing-unlocking-next-gen-wireless-performance/.

[41] Alethea. "WiCheck for wi-fi validation in test houses", Alethea Communications Technologies, Accessed: Feb. 2, 2026. [Online]. Available: https://aletheatech.com/wicheck-for-wifi-validation-in-test-houses/.

[42] QACafe. "Protocol network testing with CDRouter & NTA3000 hardware | qa | cafe", Accessed: Feb. 2, 2026. [Online]. Available: https://www.qacafe.com/test-solutions/cdrouter-benefits/cdrouter-technology.

[43] QACafe. "CDRouter wi-fi range vs rate test summaries". Section: cdrouter, Accessed: Feb. 2, 2026. [Online]. Available: https://support.qacafe.com/cdrouter/test-summary/wi-fi-range-vs-rate-test-summary/.

[44] QACafe. "CDRouter multiport test summaries". Section: cdrouter, Accessed: Feb. 2, 2026. [Online]. Available: https://support.qacafe.com/cdrouter/test-summary/multiport-test-summary/.

[45] Candela Technologies. "WiFi client emulators", Accessed: Oct. 14, 2025. [Online]. Available: https://www.candelatech.com/wifi_client_emulators.php.

[46] Candela Technologies. "CT521b-2ax4 economy LANforge WiFIRE 802.11a/b/g/n/AC/AX 2 radio WiFi traffic generator supporting 38 virtual STA interfaces", Accessed: Feb. 2, 2026. [Online]. Available: https://www.candelatech.com/ct521b-2ax4_product.php.

[47] iPerf Documentation. "iPerf - iPerf3 and iPerf2 user documentation", Accessed: Mar. 4, 2026. [Online]. Available: https://iperf.fr/iperf-doc.php.

[48] J. Malinen. "Hostapd and wpa_supplicant", Accessed: Feb. 25, 2026. [Online]. Available: https://w1.fi/.

[49] R. T. Braden, "Requirements for internet hosts - communication layers", Internet Engineering Task Force, Request for Comments RFC 1122, Oct. 1989, Num Pages: 116. DOI: 10.17487/RFC1122.

[50] P. Seite and M. Cullen, "Current practices for multiple-interface hosts", Internet Engineering Task Force, Request for Comments RFC 6419, Nov. 2011, Num Pages: 21. DOI: 10.17487/RFC6419.