An Open Source Implementation of Wi-Fi 7 Multi-Link Operation in OMNeT++

Doğanalp Ergenç and Falko Dressler
School of Electrical Engineering and Computer Science, TU Berlin, Germany
{ergenc, dressler}@ccs-labs.org

***Abstract*—IEEE 802.11be Extremely High Throughput (EHT) improves the previous Wi-Fi generations with increased connectivity, throughput, and reliability. One of its key features, multi-link operation (MLO), enables using multiple links operating on the available 2.4, 5, and 6 GHz frequency bands simultaneously for better spectrum efficiency. This also comes with a flexible orchestration of these links to address the demands of evolving wireless networks, such as low latency and high reliability. Although several studies have already analyzed the benefits of MLO with custom implementations, researchers still lack an open source platform to develop their own MLO modes and deployment settings. In this paper, we introduce an open source MLO implementation in the popular simulation toolkit OMNeT++. We also present two examples of MLO modes, namely link aggregation and redundancy, to demonstrate their effectiveness.**

***Index Terms*—802.11be, multi-link operation, Wi-Fi 7, open source, simulation, OMNeT++**

I. Introduction

IEEE 802.11be Extremely High Throughput (EHT), called Wi-Fi 7, evolves the previous standard with several new features for high throughput and real-time wireless communication [1], [2]. It enables the 6 GHz frequency band and doubles the channel width to 320 MHz, leading to a significantly increased nominal throughput [3]. The improved modulation scheme 4096-QAM increases gain by 20% compared to its predecessor 1024-QAM in Wi-Fi 6. Supporting up to 16 spatial streams in MU-MIMO, it also improves the overall spectrum efficiency [2]. Last but not least, Wi-Fi 7 introduces multi-link operation (MLO) to utilize 2.4, 5 and 6 GHz frequency bands simultaneously [4]. Compared to the aforementioned physical layer enhancements that have also gradually improved throughout several Wi-Fi generations, MLO is the critical enabler for low latency and high reliability [4], [5] in wireless networks.

MLO offers numerous benefits to fulfil the requirements of a wide range of scenarios. In augmented reality (AR) applications, for instance, it helps achieve real-time communication at high data rates for the most realistic immersive experience by taking advantage of a broader frequency spectrum [6]. Multi-connectivity over different frequency bands increases reliability, which is paramount for networking in safety-critical systems like industrial facilities [7]. Alongside its benefits, MLO raises several design considerations for its optimal use, such as synchronization of multiple links to avoid self-interference [8] and co-existence with legacy Wi-Fi devices [9].

Tackling these challenges has attracted researchers since the beginning of the 802.11be standardization efforts in 2019. Several studies in the literature present extensive analyses of the overall throughput and latency improvements in Wi-Fi 7, focusing on the MLO feature [10], [11]. As Wi-Fi 7 was only released in early 2024 and supporting devices have not been widely deployed yet, all these works simulate the partial capabilities of MLO using their custom implementation. Beyond this, there is only a limited effort on developing accessible Wi-Fi 7 and MLO tools[1]. Therefore, there is a need for an open and extensible platform that researchers can utilize and adapt according to their needs.

In this paper, we introduce an open source MLO implementation[2] for the simulation toolkit OMNeT++[12]. OMNeT++ supports several networking technologies such as IEEE 802.1 TSN protocols and 5G cellular networks [13] that Wi-Fi 7 is expected to interplay and be integrated [14], [15]. Accordingly, our implementation does not only foster the development of complete MLO features but also helps reveal its importance for evolving wireless systems. Finally, we evaluate two MLO modes using our implementation, link aggregation and redundancy, to demonstrate their throughput and reliability benefits over legacy single-link operation (SLO).

II. Background

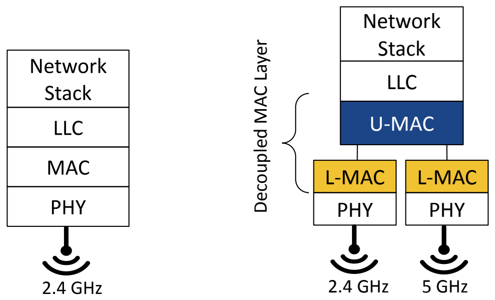
We first briefly describe the 802.11 protocol stack and its evolution for MLO support, focusing on the relevant aspects of our implementation. The protocol stack on Wi-Fi stations (STAs) and access points (APs) consists of three layers: logical link control (LLC), media access control (MAC), and physical (PHY) [16], as shown in Figure 1a. Their functions are:

1) **LLC layer** is an interface to higher layers, e.g., network layer protocols, and can optionally implement additional flow and error control functions.
2) **MAC layer** is responsible for channel access to transmit packets, avoiding interference with ongoing transmissions in a shared wireless medium. In the latest Wi-Fi standard, this layer employs hybrid coordination function (HCF) as the channel access mechanism, enabling contention-based and contention-free access. As a part of the HCF, the primary contention-based medium access function is enhanced distributed channel access (EDCA). It performs carrier sensing for collision avoidance, in which devices

[1] The event-based simulator ns-3 has announced support for the basic MLO features at https://www.nsnam.org/releases/ns-3-40/.

[2] Accessible at https://github.com/tkn-tub/wifi-mlo-omnet.

ISBN: 978-3-903176-71-3



(a) Legacy (b) MLD

Figure 1. 802.11 protocol stack.

listen to the channel and transmit when it remains free for a randomized backoff period. EDCA adapts this period for different types of traffic, i.e., categorized into access categories (AC) such as voice and video, to prioritize packets according to their quality of service (QoS) requirements.

3) **PHY layer** performs radio transmission over a certain carrier frequency.

Beyond this architecture, MLO requires differentiating various aspects of multiple interfaces, such as QoS parameters and contention mechanisms. Therefore, multi-link devices (MLDs) have decoupled upper and lower MAC layers (U-MAC and L-MAC, see Figure 1b) to orchestrate their associated links:

1) **U-MAC** constitutes a common sub-layer for all interfaces and performs link-independent functions such as frame aggregation and sequence number encoding. Outgoing frames await at U-MAC until assigned to one or multiple interfaces to be transmitted. This enables flexible link coordination (e.g., for load balancing and QoS-policy enforcement) and management (e.g., setup, association, and authentication) for better scalability and end-to-end QoS. U-MAC is also an abstraction for underlying links, rendering MLO transparent to the upper layers above MAC.
2) **L-MAC** handles link-level operations such as channel access, e.g., EDCA, aligning with the specific QoS needs and contention levels of each interface (PHY) or link. It also performs several frame-related functions, such as creating and validating MAC headers and frame integrity, similar to the unified MAC.

Lastly, depending on the functionality of U-MAC, the LLC layer can be deployed on top of it or integrated into L-MAC.

III. Implementation

We implemented MLO using the 802.11 modules in INET v4.5.2[3] framework on top of OMNeT++ v6.0.3. In OMNeT++, there are two kinds of (relevant) modules: source (C++) and network description (NED) files. Source files implement the overall logic and working principles of algorithms and network protocols. NED files define composite modules with several protocols and other modules working together. In this section, we describe the primary artifacts of our MLO design in relation to the OMNeT++ modules, parameters, and functions.

[3]INET, https://inet.omnetpp.org/

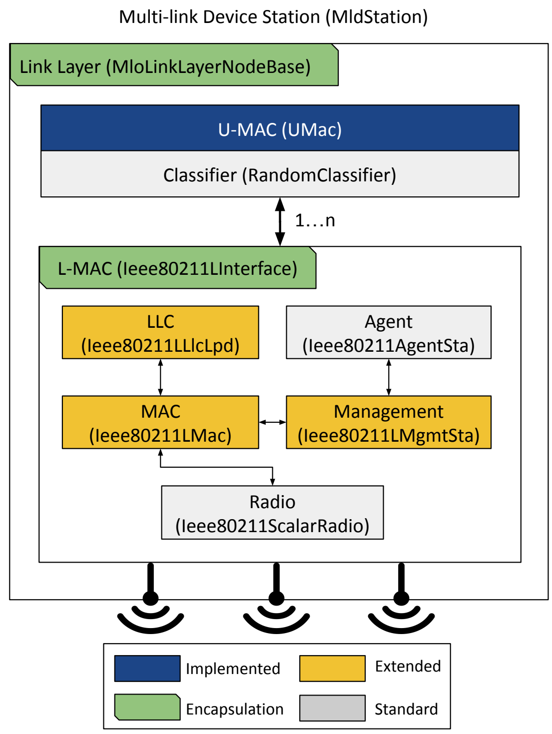


Figure 2. Overview of STA MLD.

A. General Overview

We designed two main components, `MldStation` and `MldAccessPoint`, as STA and AP MLDs, respectively. They employ the decoupled link layer architecture described in Section II. The implementation overview for STA MLD is shown in Figure 2. AP MLD has a slightly different architecture with additional management modules. In the figure, green components (link layer and L-MAC) encapsulate several modules and are implemented as NED modules. Blue components (U-MAC) are implemented from scratch, and yellow (LLC, MAC, and Management) represents extensions in built-in OMNeT++ modules. They have both source and NED files. Lastly, grey components have been directly taken from OMNeT++. The names between parentheses show the actual module names in the implementation.

The key components in Figure 2 are U-MAC (`Umac`) and L-MAC (`Ieee8021LInterface`). U-MAC can orchestrate an arbitrary number of L-MACs, i.e., set by the `numWlanInterfaces` parameter in the link layer component (`MloLinkLayerNodeBase`). In uplink, U-MAC selects suitable link(s) for data packets with different QoS classes, and L-MAC handles channel access and transmission at the selected link. In downlink, all L-MAC instances (one per link) forward data and management packets to U-MAC, which then processes all and forwards only data packets to the upper layers in the network stack.

B. Upper MAC (U-MAC)

In our implementation, U-MAC of an MLD takes role as the link orchestrator that can collect information from all associated

ISBN: 978-3-903176-71-3

links and can utilize them according to implemented MLO modes, channel conditions, and QoS characteristics of data packets[4]. It keeps a list of available radio interfaces represented by L-MAC instances. When the MLD is initialized, U-MAC assigns the same MAC address to its interfaces to render them transparent to other MLDs. This makes the origin of the uplink packets transmitted via different links perceived as the same, leading to the seamless use of multiple links.

U-MAC provides the `sendPacket` function to implement MLO modes for multi-link transmission. This function should be overridden to develop new link selection strategies. U-MAC takes the following actions before forwarding an uplink packet to one or more links:

1) It uses a packet classifier (`RandomClassifier`) to set a QoS class per packet. This is then considered for channel access functions in L-MAC.
2) It tags packets as 802.11be frames so that they can be recognized by the receiver's U-MAC module.
3) It fetches relevant parameters from channel access functions of each link, e.g., EDCA backoff counters, queue size for each AC class, via `updateChannelAccessParams` function. These parameters are beneficial to selecting suitable links leading to minimum channel access delay.

Our base U-MAC implementation (`UMac`) always selects the 5 GHz link in the uplink. We provide further examples MLO modes, whose details are given in Section IV.

In downlink, U-MAC processes all frames with 802.11be tag. For these frames, it records (`updateLinkQosParameters` function) the detected signal-to-noise and interference ratio (SNIR) as an indicator of the link quality for a better link selection decision. Lastly, it forwards the data packets to the upper layers and discards the 802.11 management frames. The latter can be used to have a better overview of the associated basic service set (BSS) to develop effective MLO modes.

C. Lower MAC (L-MAC)

To implement L-MAC, we modified the INET 802.11 interface with extended MAC (`Ieee80211LMac`), management (`Ieee80211LMgmtSta`), and LLC (`Ieee80211LLcLpd`) modules. It also contains the radio module, i.e., PHY, following a similar structure with OMNeT++.

In uplink, MAC performs channel access using EDCA with four AC classes. It accesses the extended HCF (`LHcf`) and EDCA (`LEdca` and `LEdcaf`) modules to fetch channel access parameters to be sent to U-MAC. This process involves in a chain of modules from MAC to individual EDCA functions per AC, which is shown in Figure 3.

In downlink, L-MAC places the necessary tags on the received packets. Firstly, MAC adds a link identifier for the receiving interface to assist U-MAC in distinguishing between multiple links. Secondly, LLC puts a protocol identifier for 802.11be frames. This enables interoperability to use our L-MAC module for previous Wi-Fi generations such that non-802.11be frames are directly delivered to the upper layers skipping U-MAC. Lastly, the management module is extended to forward the copies of downlink management frames to U-MAC.

[4]In an alternative design, some of the legacy MAC functions could also be implemented in U-MAC [17].

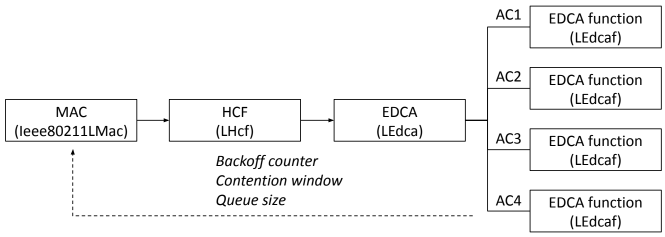


Figure 3. Fetching parameters from channel access functions.

IV. Evaluation

We evaluated our implementation in a simple scenario with two STA MLDs and one AP MLD. In the experiments, each MLD has two radio interfaces operating on 2.4 GHz and 5 GHz frequency bands, i.e., two active links for MLO, with a 20 MHz channel. Each radio interface further uses two isotropic antennas radiating equally in all directions. They have 13 dBm transmission power and −85 dBm reception sensitivity, which are aligned with typical Wi-Fi devices. We use a scalar radio medium with a fixed background noise of −110 dBm, consider free space path loss, and use the Nist error rate as the default 802.11 error model in OMNeT++.

For the data traffic, one STA sends a UDP stream to the other in the experiments. It employs EDCA as the default channel access function, and all UDP packets (1000 B) are classified as video AC. They are generated via a fixed interarrival time of 0.2 ms for 30 s. We repeat this simulation 40 times. All simulation parameters are also summarized in Table I.

For the evaluation, we implemented two modes, link aggregation (MLO-A) and redundancy (MLO-R), to illustrate different MLO use cases. They are mainly designed for demonstration and omit several MLO-specific details such as handling ACKs and managing retransmissions over multiple links, etc. In MLO-A, the U-MAC of the sender STA distributes the packets over two links randomly. This represents aggregating their

Table I
Experiment parameters.

Parameter	Value
Number of links	2 (2.4 GHz, 5 GHz
Number of antennas	2 (Isotropic)
Channel bandwidth	20 MHz
Transmission power	13 dBm
Receiving sensitivity	−85 dBm
Background noise	−110 dBm
Path loss	Free space
Error model	Nist error rate
Data traffic	UDP (Video)
Packet size	1000 B
Packet interarrival time	0.2 ms
Simulation time	30 s
Repetition	40

ISBN: 978-3-903176-71-3

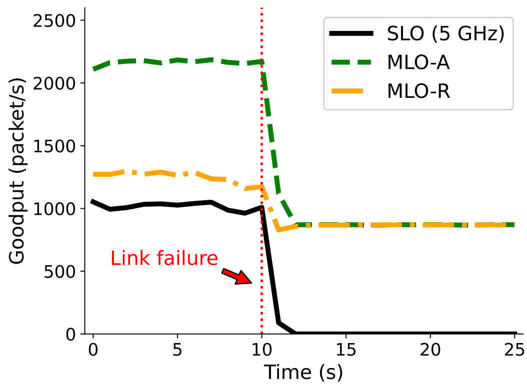


Figure 4. Goodput over time and under a link failure.

capacity under high load. In MLO-R, the U-MAC duplicates the packets on both links to tackle potential link failures for seamless fault tolerance. In the source code, these modes are implemented in `RandomUMac` and `RedundancyUMac` modules, respectively. For benchmarking, we also include single-link operation (SLO), in which an STA transmits over only one link. In the experiments, we simulated a link failure by shutting down one of the radio interfaces on the sender STA to show the effectiveness of the MLO-R in such cases. This scenario also represents jamming or high congestion cases that can cause packet losses.

Figure 4 shows the simulation results. We measure the goodput in terms of the number of packets per second, which is the rate of unique packets after eliminating the duplicate packets due to redundancy in MLO-R, over time. Note that we use this metric instead of throughput, i.e., achievable data rate, since the latter is typically low and not representative in a scenario with only two STAs. The reason is that contention-based channel access limits the transmission time per access, i.e., TXOP, due to the repeated randomized backoff procedure. In the figure, MLO-A (green, dashed) achieves the highest goodput as it utilizes both links simultaneously. MLO-R (orange, dashed, and dotted) still outperforms SLO (black, solid) since using two links redundantly helps to overcome packet losses on a link. In the simulation, this usually stems from radio queue overflow and exceeded retransmission limits under high load.

At 10 s, we shut down the sender's 5GHz link to simulate a link failure. Expectantly, SLO cannot transmit thereafter. MLO-A and MLO-R achieve the same goodput as they both utilize the second link in 2.4 GHz. However, they perform slightly worse because of the reduced data rate in the secondary link and the inability to compensate for packet losses on a single link, which MLO-R could achieve before the failure.

V. Conclusion

Wi-Fi 7 enables multi-link operation (MLO), which utilizes multiple links over different frequency bands simultaneously for increased spectrum efficiency, reliability, and overall QoS. In this paper, we introduced an open source MLO implementation in the OMNeT++ simulator. We validated and evaluated our implementation using an example scenario. Accordingly, we demonstrate two MLO modes for link aggregation and redundancy to show goodput and fault tolerance benefits of MLO under link failures. Our implementation helps researchers to design and evaluate new MLO modes regarding the optimal selection of multiple transmission links for addressing their application-specific requirements.

Acknowledgments

This work was supported by the Federal Ministry of Education and Research (BMBF, Germany) within the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K as well as by the German Research Foundation (DFG) within the projects Resilient Worlds under grant DR 639/30-1.

References

[1] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 8: Enhancements for extremely high throughput (EHT)," IEEE, Std 802.11be-D1.0, May 2021.
[2] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current Status and Directions of IEEE 802.11be, the Future Wi-Fi 7," *IEEE Access*, vol. 8, pp. 88 664–88 688, May 2020.
[3] 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. 3, pp. 12–18, Sep. 2022.
[4] A. Lopez-Raventos and B. Bellalta, "Multi-Link Operation in IEEE 802.11be WLANs," *IEEE Wireless Communications*, vol. 29, no. 4, pp. 94–100, Aug. 2022.
[5] 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 Comm. Mag.*, vol. 62, no. 8, pp. 126–132, Aug. 2024.
[6] M. Alsakati, C. Pettersson, S. Max, V. N. Moothedath, and J. Gross, "Performance of 802.11be Wi-Fi 7 with Multi-Link Operation on AR Applications," in *IEEE Wireless Communications and Networking Conference (WCNC 2023)*, Glasgow, United Kingdom: IEEE, Mar. 2023.
[7] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Multi-Connectivity as an Enabler for Reliable Low Latency Communications—An Overview," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 156–169, 2020.
[8] W. Murti and J.-H. Yun, "Multi-Link Operation with Enhanced Synchronous Channel Access in IEEE 802.11be Wireless LANs: Coexistence Issue and Solutions," *Sensors*, vol. 21, no. 23, 2021.
[9] D. Medda, A. Iossifides, P. Chatzimisios, F. José Velez, and J.-F. Wagen, "Investigating Inclusiveness and Backward Compatibility of IEEE 802.11be Multi-link Operation," in *IEEE Conference on Standards for Communications and Networking (CSCN 2022)*, Thessaloníki, Greece: IEEE, Nov. 2022, pp. 20–24.
[10] A. Lopez-Raventos and B. Bellalta, "Dynamic Traffic Allocation in IEEE 802.11be Multi-Link WLANs," *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1404–1408, Jul. 2022.
[11] T. Song and T. Kim, "Performance Analysis of Synchronous Multi-Radio Multi-Link MAC Protocols in IEEE 802.11be Extremely High Throughput WLANs," *Applied Sciences*, vol. 11, no. 1, 2021.
[12] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *European Simulation Multiconference (ESM 2001)*, Prague, Czechia, Jun. 2001.
[13] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G – An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, Jan. 2020.
[14] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-Sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7," *Sensors*, vol. 21, no. 15, p. 4954, Jul. 2021.
[15] T. Janevski, "Future Mobile and Wireless Broadband," in *Future Fixed and Mobile Broadband Internet, Clouds, and IoT/AI*, Wiley, 2024, pp. 155–192.
[16] W. Stallings, "IEEE 802.11: Wireless LANs from a to n," *IT Professional*, vol. 6, no. 5, pp. 32–37, 2004.
[17] Y. Fang, B. Sun, Z. Han, and N. Li, "Multi-link Architecture and Requirement Discussion," *Submission IEEE 802.11-2019/1095*, Jul. 2019.

ISBN: 978-3-903176-71-3