Redundancy in WiFi 7: Combining Multi-link Operation with IEEE 802.1CB FRER

Doğanalp Ergenç*, Tobias Reisinger, Falko Dressler

School of Electrical Engineering and Computer Science, TU Berlin, Germany

Abstract

The increasing complexity of modern industrial and manufacturing systems, featuring numerous sensors and mobile components, demands reliable, low-latency communication over wireless networks. WiFi 7 addresses these requirements through enhancements such as Multi-Link Operation (MLO), enabling simultaneous use of multiple frequency bands and offering inherent link diversity. This raises the question of whether MLO can be effectively leveraged for reliability in critical systems. In this paper, we explore the integration of IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER), a core Time-Sensitive Networking (TSN) standard, over MLO to address this question. We present an open-source implementation of FRER over MLO in the OMNeT++ simulator, highlight key challenges in combining these technologies, and evaluate its effectiveness in improving reliability under wireless-specific conditions such as mobility and congestion. Our results demonstrate that FRER can enhance packet delivery ratio and ensure bounded latency, albeit at the expense of reduced channel efficiency due to redundancy.

Keywords: IEEE 802.11be, WiFi 7, TSN, MLO, FRER, redundancy, reliability

1. Introduction

Modern critical systems, such as smart industrial and manufacturing facilities, consist of numerous interconnected components that demand low latency and highly reliable communication. Although traditional fieldbus and Ethernet-based networks have successfully met these requirements, the growing deployment of sensors and mobile components, such as autonomous robots, necessitates the use of wireless technologies. Consequently, ensuring reliable and near-deterministic communication over the wireless medium has become essential.

One of the most recent wireless technologies addressing these evolving needs is the IEEE 802.11be Extremely High Throughput (EHT), also known as WiFi 7. In addition to various enhancements, such as improved spectral efficiency and higher modulation and coding schemes [1], WiFi 7 introduces a key feature: multi-link operation (MLO). This feature enables WiFi stations (STAs) and access points (APs) (referred to as multi-link devices (MLDs)) to simultaneously utilize multiple frequency bands at 2.4, 5 and 6 GHz. MLO provides both frequency and spatial diversity across multiple radio interfaces, helping mitigate link failures, degraded channel conditions, and congestion on individual frequency bands. These benefits can be leveraged by dynamically switching between links or using them concurrently to achieve redundancy. However, this requires the development of intelligent, real-time link-switching strategies and efficient redundancy protocols tailored to the characteristics of wireless communication.

For designing such reliability mechanisms, we can first examine standardized approaches to evaluate their adaptability. For example, IEEE 802.1 Time-sensitive Networking (TSN) defines a set of protocols for Ethernet-based networks to ensure deterministic end-to-end communication. One of its core standards, IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER), introduces mechanisms to achieve redundancy without imposing specific topological constraints, and its effectiveness against node and link failures has been demonstrated in various studies [2]. While originally designed for multipath communication in Ethernet networks, the implementation of FRER functions is flexible to be adapted to diverse network architectures and requirements [3]. Accordingly, combining FRER with MLO offers a promising fault-tolerance mechanism that couples (i) a well-defined, configurable protocol with (ii) multi-connectivity and link diversity. This can prevent packet losses, reduce retransmissions, and improve quality of service (QoS), particularly in terms of end-to-end latency.

The coexistence of TSN and WiFi has received significant attention in recent years [4, 5]. This integration is both logical and necessary, as both technologies are standardized by IEEE and often deployed together in hybrid wired-wireless networks. However, the implications of using an Ethernet protocol for wireless communication should be considered for this integration. Although some preliminary studies have explored the use of FRER over MLO [6], the

*Corresponding author

Email addresses: ergenc@ccs-labs.org (Doğanalp Ergenç), tobias.reisinger@campus.tu-berlin.de (Tobias Reisinger), dressler@ccs-labs.org (Falko Dressler)

Article published in Elsevier Computer Communications 247 (2026) 108341. (10.1016/j.comcom.2025.108373)

effects of mobility, congestion on specific links, and challenges of FRER deployment in the wireless environment remain underexplored. Moreover, the implementation issues in extending the WiFi network stack to support FRER functionalities are not yet fully understood.

In this paper, we explore the integration principles of FRER over MLO and assess its reliability and performance. Our main contributions can be summarized as follows:

- We present an open-source implementation[1] of FRER over MLO using the simulation toolkit OMNeT++. It is based on our previous MLO implementation presented in [7].
- We detail the implementation and configuration challenges, and the necessary modifications for compatibility between these two technologies.
- We evaluate the effectiveness of FRER in addressing failures and challenges such as mobility and congestion in wireless networks.

The remainder of this paper is organized as follows: Section 2 introduces the fundamentals of FRER and MLO. Section 3 reviews related work. Section 4 gives an architectural overview of the combined FRER and WiFi stack, and Section 5 details our implementation. Section 6 presents our analysis and performance evaluation. Section 7 highlights open issues, and Section 8 concludes the paper.

2. Background

In this section, we present the fundamentals of FRER and MLO as the core features to implement redundancy over WiFi.

2.1. Frame Replication and Elimination for Reliability

IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) offers a static redundancy mechanism by replicating Ethernet frames, via multiple, preferably node-disjoint redundant paths. It also includes an elimination mechanism, running on TSN bridges, i.e., Ethernet switches in TSN terminology, and end-hosts, to discard duplicate packets. This mechanism prevents redundant transmissions across the same link and protects the network against loops and stuck senders [8].

Figure 1 shows the use of FRER in a network where three disjoint paths are assigned to a talker stream, e.g., two of them for redundancy. In the figure, either listener or the bridge that forwards two member streams can drop redundant packets. Note that a talker refers to an Ethernet host that sends data traffic, while a listener is the intended receiver of that traffic. Generally, the talker performs:

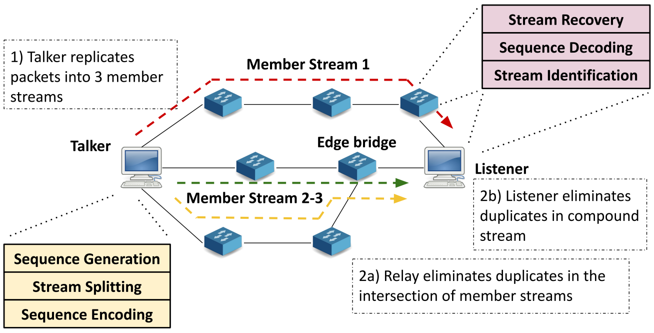

1) Talker replicates packets into 3 member streams
Member Stream 1
Stream Recovery
Sequence Decoding
Stream Identification
Talker
Edge bridge
Listener
Member Stream 2-3
2b) Listener eliminates duplicates in compound stream
Sequence Generation
Stream Splitting
Sequence Encoding
2a) Relay eliminates duplicates in the intersection of member streams

Figure 1: An example of FRER deployment with redundant paths.

- **Sequence generation** by generating a unique identifier per packet of a stream to be incremented for the other packets in the sequence.
- **Stream splitting** by copying the packets and creating member streams to be sent through k distinct paths. In this context, a member stream refers to a duplicated data stream originated from a main stream. They can exist in any number depending on the degree of redundancy, i.e., the number of redundant paths.
- **Sequence encoding** by assigning a sequence number to the copied packets via the so-called redundancy tag (R-TAG).

Note that multiple TSN bridges can exist along an end-to-end path. They can duplicate or eliminate frames on the fly according to the given configuration. On the one hand, additional duplication at intermediate bridges can protect specific network segments when necessary. On the other hand, early elimination before delivery to the listener improves link utilization by avoiding unnecessary duplicates. For instance, in Figure 1, member streams 2 and 3 can be merged on the edge bridge (i.e., the bridge directly connected to an end-host), if the access link connected to the listener is reliable.

Then, a listener or an intermediate bridge on an end-to-end path performs:

- **Stream identification** by applying a stream identification function to distinguish a stream. A mandatory *null stream identification* method uses source or destination MAC addresses and virtual local area network (VLAN) ID of a packet as input. Optionally, this function can use source and destination ports, IP addresses, and differentiated services codepoint (DSCP) as parameters.
- **Sequence decoding** by extracting the sequence identifier of a packet to be compared to the identified stream's sequence information.
- **Stream recovery** by deciding if a packet is duplicate and should be dropped or forwarded. This eventually merges member streams into the original one.

[1] The framework is available as open source at `https://github.com/tkn-tub/wifi-frer-mlo-omnet`.

- **Latent error detection** by counting if it has received the expected number of duplicate packets to detect a node or link failure on the path of a member stream.

On the receiver side, the stream recovery stage consists of two functions. The sequence recovery function (SRF) processes all the packets received from different ports of the switch, and thus, it can detect duplicate packets of a stream coming from different paths. In contrast, the individual recovery function (IRF) processes the stream coming from a single path (or port) and is effective against, for instance, duplicate packets due to a stuck sender. Any recovery function utilizes an algorithm to decide on packet forwarding or dropping. The first option is match recovery algorithm (MRA), which tracks the last sequence number received and eliminate any subsequent packets with less than or equal number. The second one is vector recovery algorithm (VRA) and it eliminates packets out of a specific sequence number window within a range. A timeout duration is set for these functions to reset the expected sequence number (and interval) to refresh the recovery function in case of not forwarding any packet for the specified duration due to occasional failures.

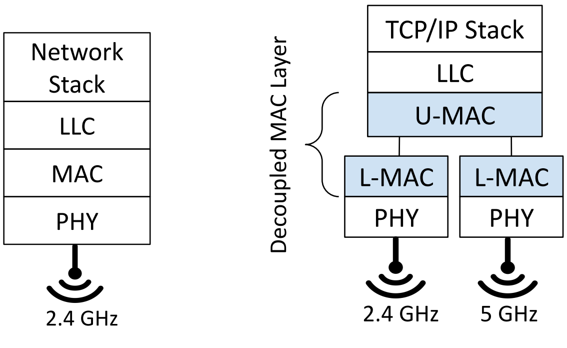
2.2. WiFi 7 Multi-link Operation

In WiFi 7, MLO eliminates the strict separation between multiple wireless interfaces deployed on STAs and APs. While WiFi 6 devices can also use 2.4 GHz or 5 GHz links selectively and one at a time, MLO-capable MLDs in WiFi terminology, utilize multiple wireless links simultaneously in coordination. This necessitates certain architectural improvements in WiFi protocol stack. Here, we briefly describe the relevant changes to our implementation.

The protocol stack on legacy WiFi devices consists of three layers: logical link control (LLC), medium access control (MAC), and physical layer (PHY) [9], as shown in Figure 2a. Their functions are:

- **LLC layer** is an interface to higher layers, e.g., network layer protocols, and can optionally implement additional flow and error control functions. It handles the multiplexing and demultiplexing of multiple upper-layer protocols over the same MAC layer.

- **MAC layer** is responsible for channel access to transmit packets, avoiding interference with ongoing transmissions in a shared wireless medium. In the latest WiFi standard, this layer typically employs hybrid coordination functions (HCF) as the channel access mechanism. As a part of the HCF, the primary contention-based medium access function is Enhanced Distributed Channel Access (EDCA) [10]. It performs carrier-sense multiple access with collision avoidance (CSMA/CA), in which devices listen to the channel and transmit when it remains free for a randomized backoff period. EDCA also adapts this period for different traffic classes (or access categories) to prioritize various traffic flows according to their QoS requirements. WiFi MAC can also leverage other channel access methods, such as orthogonal frequency-division multiple access (OFDMA) and handshake-based RTS/CTS. These methods are beyond the scope of this study, but interested readers can find more information in [11].



(a) Legacy (b) MLD

Figure 2: 802.11 protocol stack.

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

Beyond this architecture, MLO requires differentiating various aspects of multiple interfaces, such as traffic allocation, QoS parameterization, and contention mechanisms. Therefore, MLDs have decoupled upper and lower MAC layers (U-MAC and L-MAC, see Figure 2b) to orchestrate their associated links:

- **U-MAC** constitutes an abstraction 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. The abstraction of U-MAC also renders MLO transparent to the upper layers above MAC.

- **L-MAC** handles link-level operations such as channel access aligning with the specific QoS needs and contention levels at each link. It performs frame-related functions, such as creating and validating MAC headers, similar to the unified MAC. When STAs perform contention-based channel access, such as EDCA, L-MAC is responsible for channel sensing and transmission according to its QoS configuration. In contrast, for AP-assisted channel access methods (e.g., OFDMA scheduling and RTS/CTS handshakes), multiple L-MACs within a single station may require additional coordination by the U-MAC for an optimal traffic distribution and scheduling across the links.

In the WiFi 7 standard, MLO operates primarily in two modes: simultaneous transmit and receive (STR) and non-simultaneous transmit and receive (NSTR) [1]. In STR, an

MLD can independently transmit and receive over multiple links, each performing separate channel access. The data flows are assigned to links at the U-MAC through a procedure called TID-to-link mapping, which associates each flow with a specific link based on its traffic class indicated by a traffic identifier (TID). Each link's performance then depends on its individual conditions, such as noise, interference, and congestion. In contrast, NSTR requires the transmitting links of an MLD to be synchronized in order to prevent in-device coexistence (IDC) interference [12]. This type of interference occurs when simultaneous transmission and reception across multiple links causes the transmitter radios to interfere with the receiver radios within the same device. The issue arises because the transmitted signal is significantly stronger than the weak external signal the receiver is attempting to detect. Therefore, the channel access of the potentially interfering interfaces must be coordinated by the U-MAC of the corresponding MLD. Note that our implementation considers only the STR mode, in which each L-MAC of an MLD performs their own channel access for the data flows assigned by U-MAC.

3. Related Work

In this section, we present related work on the integration of FRER into wireless networks in general, and redundancy approaches in WiFi.

3.1. FRER in Wireless Networks

The use of TSN protocols, including FRER, over wireless technologies has recently been investigated in the literature. In [5], potential strategies for applying FRER over MLO are briefly discussed. The authors propose a higher-layer 802.1CB mechanism to manage redundancy tags and stream configurations, assisting the U-MAC. Adame et al. [13] highlights the IEEE 802.11ak standard, which enables architectural and protocol-level integration of FRER into MLO [14]. The standard treats 802.11 APs similarly to 802.1 TSN bridges and supports node- and link-disjoint path selection. In WiFi, this translates to link diversity enabled by MLO. These works do not include any specific implementation or evaluation but discuss FRER–MLO integration only at a conceptual level. A draft proposal in [15] builds on the latter idea, presenting a design where STAs and APs can connect to multiple peers, enabling multi-path connectivity similar to FRER. The authors do not directly suggest the deployment of FRER but discuss the benefits of redundancy over multiple APs, resembling the multi-AP coordination feature of the upcoming Wi-Fi 8 [16].

Montgomery et al. [17] combine FRER with IEC 62439 High-availability Seamless Redundancy (HSR) to provide redundancy in wireless local area networks. Although they do not directly employ MLO, their proposal utilizes WiFi components with multiple interfaces. They also demonstrate the latency benefits of FRER-based seamless redundancy in their custom prototype. In another study, Syed et al. [18] investigate FRER in vehicular networks, focusing on platooning for inter-vehicle communication. FRER has also been studied in hybrid 5G-TSN networks that combine redundancy mechanisms with 5G extensions. Works such as [19, 20, 21] examine various deployment and integration scenarios to improve end-to-end reliability across wired and wireless links. These topics lie beyond the scope of our study, which focuses specifically on FRER in WiFi 7.

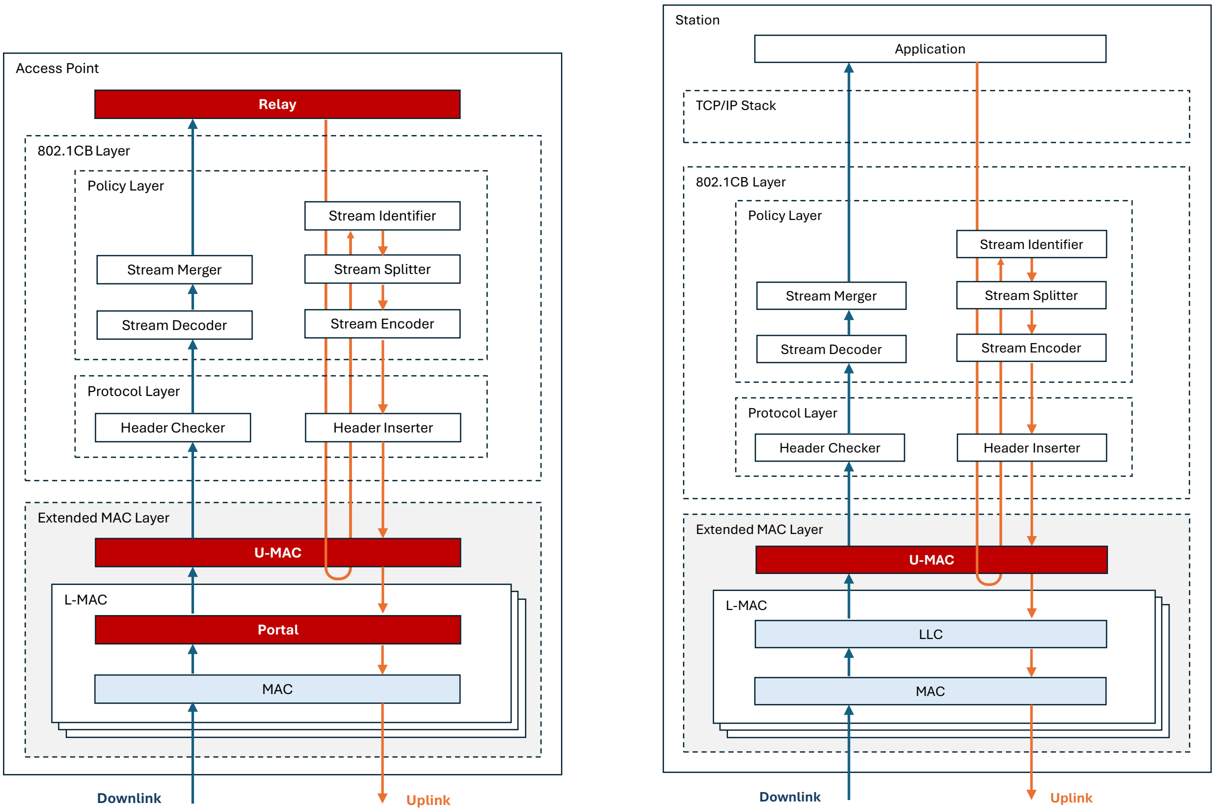
Work	**Features**		**Evaluation**	**Open-source**
	MLO	FRER		
Cavalcanti et al. [5]	✓	✓	×	×
Adame et al. [13]	✓	✓	×	×
de la Oliva et al. [15]	✓	✓	×	×
Montgomery et al. [17]	×	✓	✓	×
Syed et al. [18]	×	✓	✓	×
[19, 20, 21]	×	✓	✓	×
Suer et al. [22]	×	×	✓	×
Jauh et al. [23]	✓	×	×	×
Cena et al. [24]	✓	✓	×	×
Fang et al. [6]	✓	✓	✓	×
Ergenç and Dressler [7]	✓	×	✓	✓
This work	✓	✓	✓	✓

Table 1: Comparison of related work in terms of features, evaluation, and open-source availability of their implementation.

3.2. Redundancy over WiFi

Before the introduction of MLO, several studies [25, 26, 27] demonstrated that redundancy in WiFi improves latency and reliability. In [22], the authors propose a conditional redundancy method where duplication over multiple links is triggered only when the links are uncorrelated in terms of average latency and packet loss. They also note that this approach is most effective under low network load and when links are minimally correlated. A similar strategy is presented in Jauh et al. [23], where conditional duplication is applied to real-time traffic using MLO. Duplication is triggered when the per-packet queueing delay exceeds a predefined threshold. For instance, with a 40% threshold and a 10 ms latency budget, an MLD duplicates packets after a 4 ms queueing delay. Lower thresholds increase duplication frequency, improving reliability at the cost of spectral and energy efficiency. None of these studies explicitly leverage FRER or investigate its applicability in a more general time-sensitive WiFi setting.

In [24], the authors propose a conceptual design for integrating FRER over WiFi in both single- and multi-AP configurations. They discuss several challenges in achieving seamless redundancy in WiFi. While both MLO and FRER are part of their proposal, the authors do not conduct any experiments evaluating the performance of an integrated MLO–FRER framework. Fang et al. [6] present a prototype implementation of FRER using proprietary WiFi interfaces. Instead of MLO, they use two separate sender devices and two interfaces on one listener device. Their evaluation shows that FRER over WiFi can achieve 5 ms bounded



(a) AP MLD

(b) STA MLD

Figure 3: Extended 802.11 protocol stack for IEEE 802.1CB support over MLO.

latency with 99.9% reliability. However, their proposal and evaluation are tightly dependent on their custom MLO prototype built on proprietary equipment.

3.3. Our Contribution

Table 1 compares the aforementioned works in terms of (i) the considered features, namely MLO and FRER, (ii) whether an evaluation was conducted, and (iii) the open-source availability of their proposed frameworks. These studies are generally conducted using hardware and software platforms specifically tailored for experimental purposes. Alternatively, some use custom analytical models or simulation tools. In contrast, we aim to provide an open-source implementation of the combined FRER and MLO stack within the OMNeT++ network simulator. This ensures the reproducibility of our experiments and enables the research community to conduct further analyses on various TSN-WiFi deployments. Unlike conceptual studies that explore the main design concerns for enabling FRER over wireless technologies, we present and discuss the implementation details and potential caveats of integrating FRER with FRER. For this, we use our previous MLO implementation [7] and extend it with FRER functions in OMNeT++. Although we incorporate various design insights and recommendations from [5, 24], our implementation adheres to OMNeT++ coding conventions to ensure compatibility with other protocols supported by the simulator. This approach enables us to explore design aspects relevant to a practical implementation of the combined stack, while maintaining simplicity by omitting additional standardization elements defined in [14]. Finally, we evaluate the performance of FRER-enabled redundancy in congestion and mobility scenarios, which have received limited attention in prior work.

4. Design Overview of FRER-MLO Integration

In our design, we combine an extended MAC stack with FRER functions. While AP and STA MLDs (implemented as `Ieee8021CbAccessPoint` and `Ieee8021CbStation`, respectively) share a largely similar architecture, they differ in a few aspects, as illustrated in Figure 3. In the figure, components in red and notated with bold text are either modified or newly implemented to extend the existing WiFi modules in OMNeT++ with FRER-awareness. The light blue components (LLC and MAC) are reused from our previous work [7]. All other modules come from the simulator, with minor configuration changes to integrate them into the extended stack.

The extended MAC stack consists of U-MAC and L-MAC. Their functionalities are briefly described in Section 2.2, with implementation details provided in [7]. In our design, the AP MLD includes an additional relay module that bridges multiple interfaces based on forwarding

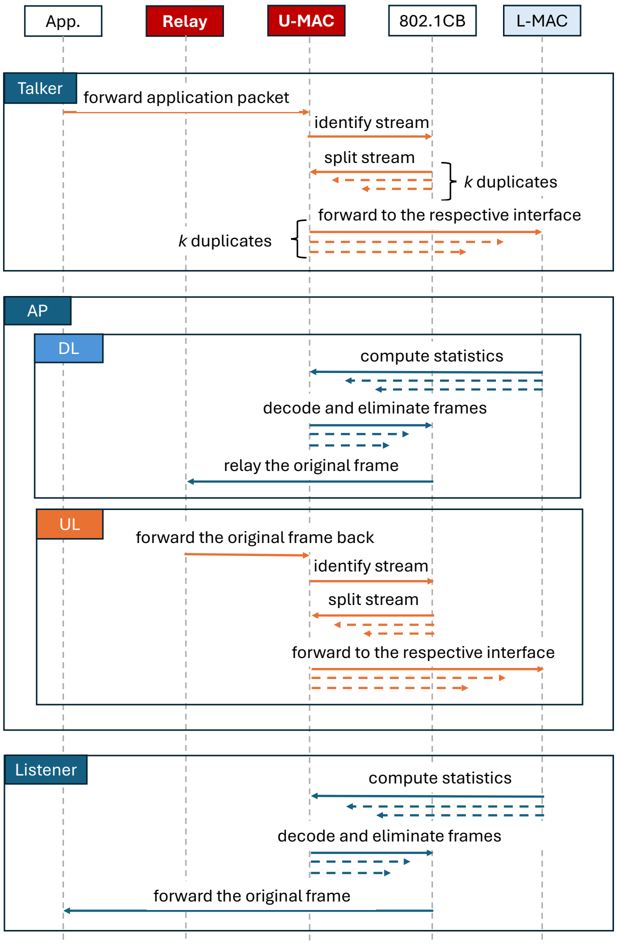

App.
Relay
U-MAC
802.1CB
L-MAC
Talker
forward application packet
identify stream
split stream
k duplicates
forward to the respective interface
k duplicates
AP
DL
compute statistics
decode and eliminate frames
relay the original frame
UL
forward the original frame back
identify stream
split stream
forward to the respective interface
Listener
compute statistics
decode and eliminate frames
forward the original frame

Figure 4: Sequence diagram for FRER-enabled communication.

rules and protocol specifications. For 802.11be interfaces, this function is primarily handled by the U-MAC.

The FRER functions are implemented as a standalone 802.1CB layer. Through uplink (UL) and downlink (DL), the stack invokes different functions in a certain sequence, as outlined in Section 2.1. We deliberately avoided modifying the 802.1CB layer to minimize changes to the simulator, addressing compatibility issues instead in the surrounding (red, bold) modules.

Based on the components shown in Figure 3, Figure 4 illustrates a simplified frame processing sequence for communication between talker and listener STAs, and an AP.

On the talker side, the application layer (i.e., any layer above MAC) generates packets and forwards them to the U-MAC. These packets are then passed to the 802.1CB module for stream identification. If they belong to an FRER-enabled stream, they are duplicated into k member streams and encoded. The resulting packets are returned to the U-MAC and forwarded to the L-MAC interfaces for transmission.

When FRER frames are received by the AP, they are first sent to the U-MAC for computing relevant metrics and statistics. The frames are then decoded into member streams and merged into the original stream by the 802.1CB module. The resulting frame is passed to the relay, which forwards it to the appropriate next layer, typically the U-MAC for 802.11be frames. For UL transmission, the same process as on the talker side is repeated. Notably, the member streams are merged during DL and then split again for UL. This allows for flexible FRER configurations at different wireless nodes. For example, even if the talker STA uses only two links, the AP can scale the redundancy to all available interfaces once it receives the original stream.

At the listener side, processing mirrors the DL flow at the AP, except that the merged frames are delivered to the application layer instead of being relayed further.

With this high-level architecture of MLDs and the overall FRER-enabled MLO flow established, we detail the implementation of each module in the next section.

5. Implementation

In our previous work, we implemented MLO using the 802.11 modules in the INET v4.5.2[2] framework on top of OMNeT++ v6.0.3 [7]. In this work, we extend that implementation by integrating existing FRER functions available in the same framework.

OMNeT++ distinguishes between two types of modules: source files (C++) and network description (NED) files. Source files define the core logic and behavior of algorithms and protocols, while NED files describe complex modules composed of various protocol components and submodules. To support information exchange beyond control and data packets, OMNeT++ also employs *tags*. These are internal annotations accessible across modules but do not appear in actual network traffic.

This section outlines the primary artifacts of our integrated FRER-MLO design, detailing the relevant OMNeT++ modules, configuration parameters, and their functional roles, as shown in Table 2. Some components are purely composite (i.e., defined only via NED files), while others involve additional functionality to support MLO or FRER features. The table also indicates whether each component is used at the AP, STA, or both.

5.1. FRER Functions

We begin by presenting the FRER functions, as their configurations are also relevant to other components in the protocol stack. These functions are encapsulated within a single IEEE 802.1CB layer (see Figure 3), implemented as the compound module `Ieee8021rLayer`. This layer is organized into two main sublayers: the protocol layer and the policy layer.

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

Component		Module	Type	Feature		Node	
				MLO	FRER	AP	STA
Extended MAC	**L-MAC**	`Ieee8021LInterface`	NED	✓	✓	✓	
	Portal	`Ieee8021CbPortal`	NED, source	✓	✓	✓	
	LLC	`Ieee80211LLlcLpd`	NED, source	✓			✓
	MAC	`Ieee8021LMAC`	NED, source	✓		✓	✓
	U-MAC	`Ieee8021CbUMac`	NED, source	✓	✓	✓	✓
FRER	**802.1CB Layer**	`Ieee8021rLayer`	NED		✓	✓	✓
	Protocol Layer	`Ieee8021rProtocol`	NED		✓	✓	✓
	Header Checker	`Ieee8021rTagEpdHeaderChecker`	NED, source		✓	✓	✓
	Header Inserter	`Ieee8021rTagEpdHeaderInserter`	NED, source		✓	✓	✓
	Policy Layer	`StreamPolicyLayer`	NED		✓	✓	✓
	Stream Decoder	`StreamDecoder`	NED, source		✓	✓	✓
	Stream Encoder	`StreamEncoder`	NED, source		✓	✓	✓
	Stream Merger	`StreamMerger`	NED, source		✓	✓	✓
	Stream Splitter	`StreamSplitter`	NED, source		✓	✓	✓
	Stream Identifier	`StreamIdentifier`	NED, source		✓	✓	✓
Relay		`Ieee8021CbRelay`	NED, source		✓	✓	

Table 2: System components, their implementation modules, types, relevant features, and the network stacks they take part.

5.1.1. Protocol Layer

The protocol layer (`Ieee8021rProtocol`[3]) provides the functions for inserting and parsing R-TAGs in FRER frames. It includes two main components: a header inserter and a header checker. They require no additional configuration and can be used with their default settings.

Header Inserter. This component (`Ieee8021rTagEpdHeaderInserter`) creates and injects R-TAGs into UL frames and encodes the sequence number, implementing the sequence encoding function of FRER. R-TAGs are defined as `Ieee8021rTagEpdHeader` and inserted at the beginning of the packet. The header inserter embeds the sequence number provided by the policy layer (see Section 5.1.2). This number is carried via the tag `SequenceNumberReq`.

Header Checker. DL frames are first passed to the header checker (`Ieee8021rTagEpdHeaderChecker`) in the 802.1CB layer to read the embedded sequence number from the R-TAG. The header checker only performs sequence number extraction and does not handle recovery. Instead, it generates a new tag, `SequenceNumberInd`, and forwards the decoded number to subsequent FRER components. Simultaneously, it removes the R-TAG from the frame.

5.1.2. Policy Layer

The policy layer (`StreamPolicyLayer`) encompasses the core FRER functions: stream identification, sequence generation, stream splitting, and sequence recovery. It includes several other submodules performing the aforementioned functions. Note that these functions are also grouped under further compound modules, which are omitted in the following description for simplicity.

Stream Identifier. This module handles UL frames for stream identification and sequence number generation. It is implemented in the `StreamIdentifier` module. In the simulator, stream identification does not strictly follow the standard FRER method, which typically uses source and/or destination MAC addresses and VLAN ID. Instead, it supports flexible matching expressions that can include IP fields, TCP/UDP ports, and other parameters to associate packets with streams. When a packet matches the configured expression, a sequence generation function assigns the appropriate sequence number. If the stream is already known, the sequence number is incremented; otherwise, it starts from 0. The assigned sequence number is then encapsulated in a `SequenceNumberReq` tag, which will later be converted into an R-TAG. If a packet already contains a `StreamReq` tag (e.g., a DL packet relayed via the AP), it is forwarded without modifying the existing sequence number. The module must be configured with a list of streams, each defined by a stream name (`stream`) and a matching expression (`packetFilter`). An example of NED configuration is shown below. In this case, the module labels UDP packets with a destination port of 5000 as belonging to *stream1* and enables sequence numbering for consecutive packets. To disable FRER on a node, it is sufficient to omit the stream mapping in the `StreamIdentifier` module.

```
**.identifier.mapping = [{
    stream: "stream1",
    packetFilter: expr(udp.destPort ==
        5000),
    sequenceNumbering: true }]
```

Stream Splitter. In the UL, identified stream packets are forwarded to the stream splitter (`StreamSplitter`), which

[3] This name refers to the R-TAG and does not denote a protocol.

determines the redundancy level. It creates multiple member streams (see Section 2.1) by replicating the original application packets according to the configured redundancy. Stream splitting, and thus frame replication, requires configuration in the NED file of the stream identifier, as shown below. For example, packets matched to *stream1* are replicated twice into member streams *stream1.1* and *stream1.2*. The original packet is discarded and packets without a `StreamReq` tag are ignored.

```
**.splitter.mapping = {
    stream1: ["stream1.1", "stream1.2"]}
```

Stream Encoder. Replicated packets are passed through the stream encoder (`StreamEncoder`). Despite its name, this component does not perform *sequence* encoding but ensures packets include a VLAN ID and a priority code point (PCP), based on its configuration. The stream encoder is not actively used in our design since VLAN and PCP values are not strictly required for WiFi, unlike other TSN protocols. Instead, the header inserter handles sequence number encoding as specified by the stream identifier. Consequently, `StreamEncoder` remains unconfigured in our experiments, but can be enabled in VLAN-enabled setups.

Stream Decoder. In the DL, received frames are first processed by the stream decoder (`StreamDecoder`). Similar to the stream identifier in the UL, it performs stream identification but with fewer parameters. This includes checking source and destination MAC addresses, VLAN and PCP values, and the receiving interface, which is critical for MLO. The module matches received frames to member streams based on the interfaces assigned to these streams. In the following configuration, the module is configured to tag any packet received on interface *mlo-mac1* with the stream name *stream1.1* when matching a given destination address. This member stream is then merged into the main stream by the stream merger.

```
**.decoder.mapping = [{
    stream: "stream1.1",
    interface: "mlo-lmac1",
    destination: "0A-AA-00-00-00-03"}]
```

Stream Merger. Lastly, the stream merger (`StreamMerger`) performs sequence recovery to eliminate duplicate packets. It provides a SRF similar to VRA, which stores received sequence numbers for each main stream (not member streams). By default, it tracks the last 10 frames. Each incoming frame is checked against stored sequence numbers; duplicates are dropped, while new sequence numbers are forwarded to the application and update the SRF buffer. The stream merger should be configured to merge member streams into the main stream. The identifiers of the member streams, e.g., names in string format, are assigned by the stream decoder as described above. The SRF is performed after the main stream is constructed. An example of configuration for merging *stream1.1* and *stream1.2* into *stream1* is shown below.

```
**.merger.mapping = {
    stream1.1: "stream1",
    stream1.2: "stream1" }
```

To remove the R-TAG after elimination, the `StreamMerger` module must be configured with empty target streams (*stream1*) at the listener.

5.2. Extended MAC Layer

In the extended MAC layer, we implement a new FRER-capable U-MAC component, while L-MAC is mostly reused from [7] except small modifications in LLC.

5.2.1. U-MAC

The U-MAC (`Ieee8021CbUMac`) orchestrates the use of underlying links to forward replicated packets based on the configured redundancy. This requires flexible stream-to-interface mapping, enabling any of the $\binom{n}{k}$ combinations across n available links on a STA or AP MLD. Additionally, U-MAC must be configured across intermediary wired or wireless bridges, when re-splitting streams is necessary. Other functions than replication, such as frame elimination, sequence recovery, and stream merging, are fully handled by the 802.1CB layer.

The key artifact in U-MAC is the new MLO forwarding table. Unlike a traditional MAC forwarding table, which maps destination MAC addresses to interfaces, our FRER-aware forwarding table (`forwardingTable`) maps streams to specific WiFi interfaces[4]. This allows U-MAC to complement stream splitting by forwarding member streams to their designated interfaces. For example, the following configuration sends frames of *stream1.1* to L-MAC *mlo-lmac1*, and those of *stream1.2* to *mlo-lmac2*. Note that stream-to-interface assignment could also be directly implemented in FRER modules but we aim to use FRER functions independent of the underlying link layer logic as much as possible.

```
**.umac.forwardingTable = {
    stream1.1: "mlo-lmac1",
    stream1.2: "mlo-lmac2"}
```

The FRER-enabled U-MAC introduces a modified packet processing logic compared to the original U-MAC described in [3]. When an STA sends a packet, U-MAC first checks for the presence of an R-TAG inserted by the 802.1CB layer. If no such header is found, the packet is forwarded back to the stream identifier for processing (see Section 2.2, orange line at UL). The packet then passes through the relevant FRER functions in sequence and returns to U-MAC only

[4] While this mapping resembles with traffic identifier (TID)-to-link mapping in WiFi, our table is specifically designed for the member streams of FRER traffic. This necessitates additional packet processing logic in U-MAC.

if it is associated with a stream. In this case, it carries an R-TAG header along with a `StreamReq` tag. Packets associated with FRER member streams are forwarded to their designated interfaces, as configured in the new forwarding table. Packets not belonging to any stream have their R-TAG stripped and are sent over the default network interface configured in U-MAC.

APs also check the frames for the presence of an R-TAG, and `StreamInd` and `SequenceNumberInd` tags (see Section 2.2). These tags exist in the UL frames if they are received from an FRER-enabled STA before. They are then converted to their respective counterparts, `StreamReq` and `SequenceNumberReq`, and forwarded back to the 802.1CB layer for stream splitting and encoding. Similar to the UL transmission in STA, the frames are finally forwarded to their respective interfaces according to the configuration of the forwarding table at U-MAC. For non-FRER frames, the AP forwards the packets through the interfaces on which they were received, following the baseline MLO logic described in [7].

5.2.2. L-MAC

In the L-MAC (`Ieee8021LInterface`), we modify only the LLC submodule on the AP side, referred to as the portal (`Ieee80211CbPortal`). This modification ensures compatibility for coexisting FRER and non-FRER traffic. Specifically, the modified portal reorders the Ethernet MAC header and the R-TAG to ensure that the R-TAG is correctly processed and removed by the 802.1CB layer when necessary. The MAC module itself (`Ieee8021LMAC`), along with its subcomponents responsible for channel access and contention, remains unchanged from the original 802.11 implementation in INET.

It is important to note that the default LLC layer in the WiFi stack performs LLC protocol discrimination (LPD) encoding. However, FRER frames are distinguished by their EtherType field, making EtherType protocol discrimination (EPD) encoding a more appropriate choice. Our portal modification facilitates the use of EPD-style headers (e.g., the R-TAG) within the LPD-based LLC environment. As a result, this design maintains compatibility with other OMNeT++ projects that rely on default configurations. For a detailed discussion on EPD/LPD differences, we refer interested readers to the presentation by the IEEE 802.1 Maintenance Task Group [28].

5.3. Relay

Finally, in the AP, the relay module (`Ieee8021CbRelay`) directs received frames to the appropriate outgoing interfaces. In our design, we do not use the native relaying functionality of this module, as interface selection is fully handled by the U-MAC. Since the relay originally removes all tags and headers that do not typically take place in WiFi frames, we also modified this module to preserve FRER-specific tags, namely `StreamInd` and `SequenceNumberInd`.

Parameter	**Value**
#MLO links	2 (2.4 GHz, 5 GHz)
Bitrate	52 Mbit/s (2.4 GHz) 130 Mbit/s (5 GHz)
Channel bandwidth	20 MHz
Modulation scheme	64-QAM
#Antennas per link	2 (Isotropic)
Transmission power	13 dBm
Receiving sensitivity	−85 dBm
Background noise	−110 dBm
Path loss	Log normal shadowing ($n = 2$)
Error model	Nist error rate
#STAs	2-24
Area	60 m × 60 m
Mobility model	Random waypoint (1–9 m/s)
Packet size	1000 Byte
Duration	15 s
Repetitions	10

Table 3: Simulation parameters.

6. Evaluation

In this section, we present our evaluation results for the analysis of FRER-MLO integration and its benefits in congestion and mobility scenarios.

6.1. Experimental Setup

We conduct the evaluation on a laptop with 16 CPUs (2.7 GHz) and 32 GB of memory. As specified in Section 5, we use OMNeT++ v6.0.3 and INET v4.5.2, in which our framework is implemented. In all experiments, we use MLDs with two interfaces operating at 2.4 GHz and 5 GHz[5]. Each interface is configured with a different bitrate, 52 Mbit/s and 130 Mbit/s, to assess the impact of asymmetrical links on redundancy. For channel modeling, we primarily use the simulator's default parameters. All relevant configurations are listed in Table 3.

Our network model consists of up to 24 stationary and mobile STAs, along with an AP, within a 60 m × 60 m area. Mobile nodes follow the random waypoint mobility model at various speeds. We use a UDP application to generate traffic, sending 1000 Byte packets for 10 s. The application generates packets at a fixed rate to fully utilize the 2.4 GHz link with a bitrate of 52 Mbit/s, ensuring that MLDs do not experience frame drops due to limited queue size. This results in an effective load of approximately 30 Mbit/s under the given simulation parameters, mainly due to channel access overhead such as backoff times, interframe spacing, and ACK frames. When multiple sender STAs are present, each runs an individual UDP application. In such cases, the 30 Mbit/s load is divided equally among them; for example, with ten STAs, each transmits

[5] At the time of writing, OMNeT++ does not support 6 GHz links envisioned for WiFi 7.

at a constant rate of 3 Mbit/s. In all evaluation scenarios, MLDs perform contention-based EDCA independently on each link. Each experiment lasts 15 s, with UDP traffic transmitted between $t = [1\text{s}, 11\text{s}]$.

The results for preliminary analysis and validation in Section 6.4 are based on single runs without repetition. For all remaining experiments, we report the average of 10 runs with nodes randomly distributed within the area.

6.2. Scenarios

In our evaluation, we consider seven scenarios, as listed in Table 4. Firstly, in Scenario 0, we validate our implementation by verifying its compliance with the protocol description given in Section 4 and illustrated in Figure 3. In Scenarios 1–4, we analyze the behavior and evaluate the performance of FRER-MLO integration under various failure conditions in a simple topology, shown in Figure 5. These scenarios cover failures of the 2.4 GHz and 5 GHz interfaces, occurring either at the sender STA or the AP, independently. In Scenario 5, we examine the benefits of redundancy under congestion scenarios involving multiple STAs. Finally, in Scenario 6, we evaluate its performance under mobility conditions. In Scenarios 1–4, a single sender transmits UDP traffic that fully utilizes the 2.4 GHz link. In Scenarios 5 and 6, where multiple stations are present, each sender transmits a proportional load to avoid channel congestion, as described in the previous section. The configuration files for all scenarios are also provided with our open-source implementation.

In these scenarios, we deploy a mix of single-link operation (SLO) STAs and MLDs. In SLO, STAs can use only one link, either at 2.4 GHz or 5 GHz. For MLDs, we employ two modes of MLO:

- **FRER mode** transmits frames redundantly over the configured links, as described in Section 5.
- **Link aggregation mode** distributes frames uniformly across the links. This improves channel efficiency but lacks the reliability benefits of redundancy.

Lastly, we report the following performance metrics:

- **Goodput** refers to the number of *unique* packets delivered per second. We use this instead of throughput to account for redundancy overhead.
- **Packet delivery ratio** is the ratio of packets received at the destination to those originally generated at the source. This is measured at the application layer, excluding redundant frames and retransmissions.
- **Latency** is measured end-to-end, from the sender's U-MAC to the receiver's U-MAC. It includes queuing and channel access delays at the L-MAC, as well as propagation delay over the wireless medium.

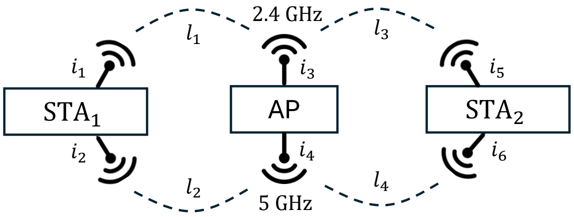


Figure 5: Base topology considered in Scenario 0-4.

Scenario	#STAs	Failure	Mobility
Scenario 0	2	×	×
Scenario 1	2	i_1 at STA$_1$ (2.4 GHz)	×
Scenario 2	2	i_2 at STA$_1$ (5 GHz)	×
Scenario 3	2	i_3 at AP (2.4 GHz)	×
Scenario 4	2	i_4 at AP (5 GHz)	×
Scenario 5	2-24	×	×
Scenario 6	16	×	✓

Table 4: Simulation scenarios. The components at the failure column refer to Figure 5.

- **Jitter** measures the average latency difference between consecutive frames. It represents the variation in latency over time and is expected to remain low and stable in reliable communication.
- **Normalized Latency Reduction (NLR)** quantifies the latency improvement achieved per unit of redundancy overhead in FRER, compared to SLO. It expresses the reduction in latency (in milliseconds) for every 1% of redundant traffic and is calculated as follows:

$$\text{NLR} = \frac{L_{\text{SLO}} - L_{\text{FRER}}}{\left(\frac{N_{\text{FRER}} - N_{\text{SLO}}}{N_{\text{FRER}}}\right) \times 100} \qquad (1)$$

  In the formula, L_{SLO} and L_{FRER} denote the average latency of the sender SLO STAs and FRER-enabled MLDs measured in a given experiment, respectively. Similarly, N_{SLO} and N_{FRER} represent the number of delivered frames in the respective experiments, with the latter also including duplicate frames.

Note that NLR can be calculated using different latency metrics, such as mean or maximum latency, and jitter, as presented in the following sections. Normalization by redundancy overhead is applied to make NLR independent of the absolute number of packets, which depends directly on the experiment duration and traffic model.

6.3. Validation Scenario

First, we present a validation scenario that illustrates the steps for the end-to-end delivery of a single UDP packet between a sender and a receiver, following the protocol description in Section 4. These steps occur as *events* in the discrete event-based simulator OMNeT++, where each

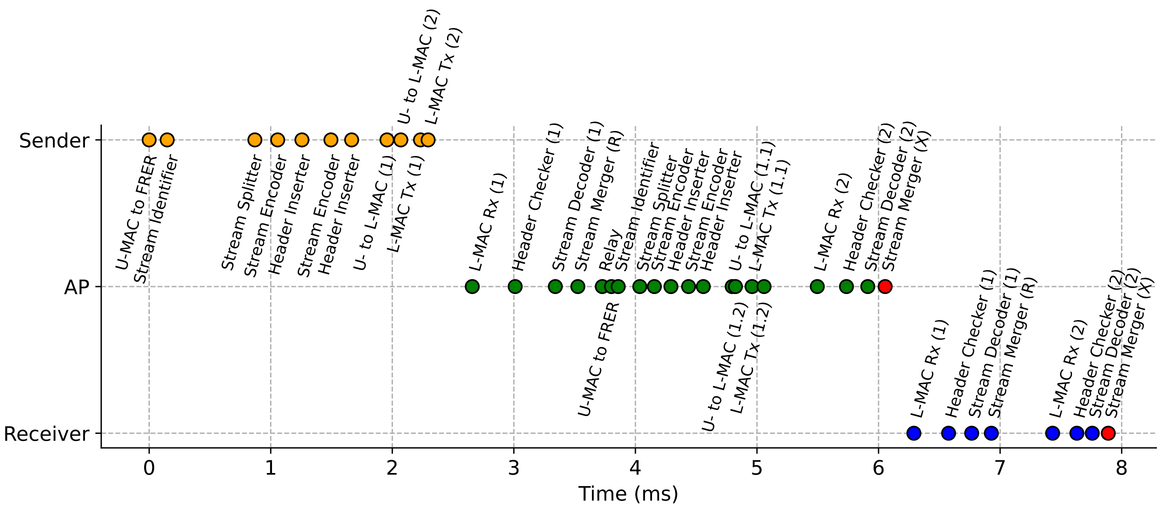


Figure 6: Scenario 0 – Validation.

event, such as stream identification, sequence number encoding and frame duplication, is scheduled in *simulation time* and executed in *real* time. Figure 6 shows these events (dots) occurring at the sender (top, yellow), AP (middle, green), and receiver (bottom, blue) in real time, annotated with the event names and relevant components.

At the sender, U-MAC forwards the application packet to the FRER module for stream identification and splitting. Two duplicates then pass through stream encoding and header insertion. After attaching the R-TAGs, U-MAC sends them to the L-MACs of two interfaces, shown as `U- to L-MAC (1/2)` events in the figure. At the AP, frame processing begins upon reception. After receiving the first duplicate (`L-MAC Rx (1)`), the frame undergoes header checking, stream decoding, and merging to verify if another frame with the same sequence number has already arrived. If not, the frame passes through the AP's *Relay* and is forwarded to U-MAC. It is then duplicated again and transmitted via the AP's L-MACs (`L-MAC Tx (1/2)`). When the second duplicate arrives (`L-MAC Rx (2)`), it is discarded by the stream merger, shown as a red dot (`Stream Merger (X)`). Finally, at the receiver, the first duplicate is retained (`Stream Merger (R)`) while the second is eliminated (`Stream Merger (X)`, red dot). Overall, this timeline corresponds directly to the communication sequence in Figure 4 and confirms the correct behavior of our implementation.

6.4. Failure Scenarios

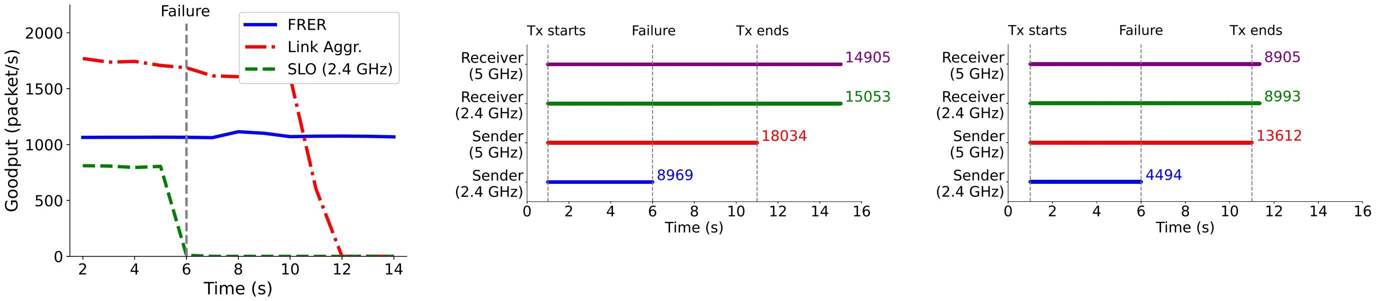
This section presents the results of the first four scenarios. Our primary objective is to analyze the behavior of wireless FRER under failure conditions. In all scenarios, we use the topology illustrated in Figure 5. STA_1 and STA_2 are designated as the sender and receiver, respectively. In each scenario, a different interface failure is injected at 6 s.

Scenario 1. Figure 7 illustrates the analysis of Scenario 1, in which i_1 (2.4 GHz) of STA_1 (i.e., the slower interface of this STA) fails spontaneously at 6 s. Specifically, Figure 7a shows the goodput over time for SLO and the different MLO modes. In that figure, the FRER mode (blue, solid) maintains stable performance after the link failure but has a lower goodput due to redundancy overhead. Nevertheless, it outperforms SLO (green, dashed) since redundancy compensates for occasional latent or erroneous frames. The link aggregation mode (red, dotted-dashed) achieves the highest goodput by utilizing both 2.4 GHz and 5 GHz links simultaneously. It experiences only a slight drop after the failure. However, the capacity of the remaining 5 GHz link is sufficient to carry all the traffic, which then continues to be distributed over both links at the AP until STA_1 stops transmission at $t = 11$ s. In contrast, SLO drops to zero immediately after the only available interface of STA_1 fails. Finally, the FRER mode continues transmitting longer (i.e., a longer blue and solid line) because it sends approximately twice as many frames. These frames are queued at the L-MAC and transmitted after channel contention.

For a more detailed analysis, Figure 7b and 7c show the frame transmission (Sender 2.4 GHz and 5 GHz, blue and red lines) and reception (Receiver 2.4 GHz and 5 GHz, green and purple lines) timelines for the FRER and link aggregation modes, respectively.[6]

At the end of each line, we also annotate the number of frames sent or received during the corresponding time interval. In both figures, transmissions on one interface (Sender 5 GHz, blue) stop after the failure, and continue only on the other (Sender 2.4 GHz, red). Although the AP receives only one copy of each UDP packet from STA_1 after the failure, it still replicates the packets into two member streams for fault tolerance across l_3 and l_4 to STA_2. As a result, the receiver in FRER mode receives significantly

[6] While these results are depicted as timelines, they actually represent the individual transmission and reception timestamps of successive frames with short interarrival times on the order of milliseconds. Thus, the disconnections within the lines result from missing frames.

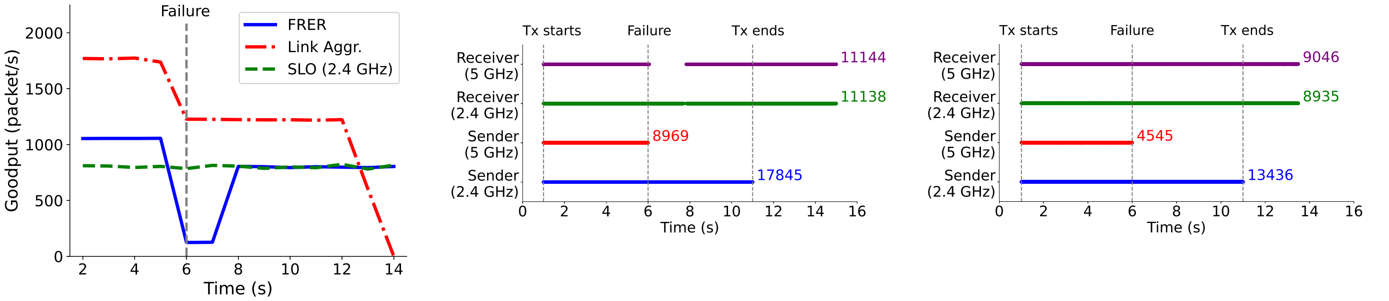


(a) Goodput over time.

(b) Tx and Rx timeline in FRER mode.

(c) Tx and Rx timeline in link aggregation mode.

Figure 7: Scenario 1 – Failure of the sender's interface at 2.4 GHz (i_1 at STA$_1$).



(a) Goodput over time.

(b) Tx and Rx timeline in FRER mode.

(c) Tx and Rx timeline in link aggregation mode.

Figure 8: Scenario 2 – Failure of the sender's interface at 5 GHz (i_2 at STA$_1$).

more frames than in link aggregation mode, as observed in the total number of frames at the links of Receiver 5 GHz and 2.4 GHz, (purple and green lines).

In conclusion, Scenario 1 confirms the stable performance of FRER under a basic link failure scenario, as well as its overhead when compared to link aggregation, which prioritizes efficiency over redundancy.

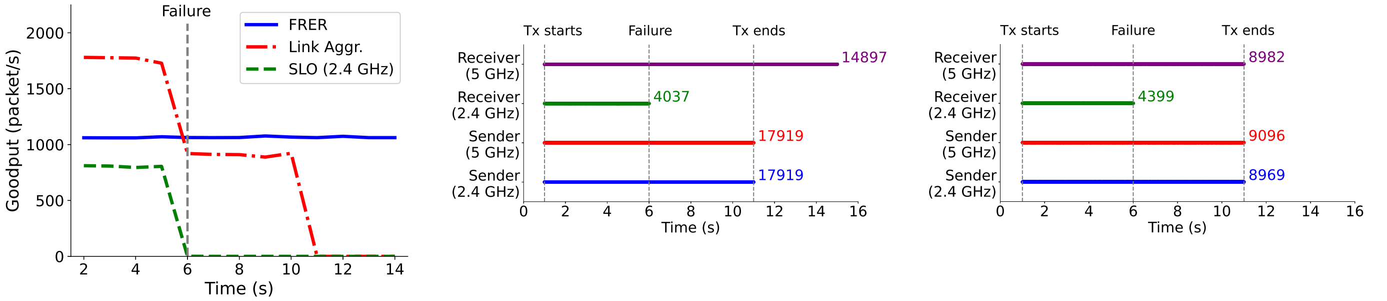
Scenario 2. Figure 8 presents the analysis of Scenario 2, where i_2 (5 GHz) of STA$_1$ (i.e., the faster interface of this STA) fails. The results are presented in the same format as Scenario 1. In Figure 8a, the link aggregation mode shows a significant decrease in goodput compared to Scenario 1, as it can no longer utilize the faster link. Consequently, frames accumulate in the queue of the L-MAC at 2.4 GHz link. SLO remains unaffected by the failure since it transmits solely over the 2.4 GHz link. In the FRER mode, we observe a sudden drop in goodput following the failure, which breaks the expectation of seamless redundancy. This is caused by an interesting phenomenon previously discussed in [2], explained below.

When FRER operates over non-disjoint and asymmetrical paths, such as links with different speeds connected to the same AP, frames transmitted over the faster link consistently arrive at the junction node (the AP) first. As a result, duplicate frames arriving later from the slower link are discarded. If the faster link fails, frames from the slower link will not be forwarded until their sequence numbers catch up with the last sequence received on the now-failed faster link. This behavior is clearly visible in Figure 8b. Between $t = [6\,\text{s}, 7\,\text{s}]$, the AP drops all packets sent by STA$_1$ over the slower link (Sender 2.4 GHz, blue); thus, no new frames appear on the receiver's timeline at 5 GHz (purple). During this time, only the frames already present in the AP's transmission queue are forwarded to STA$_2$. This explains the low yet non-zero goodput observed in Figure 8a, and the continuity in reception at the Receiver 2.4 GHz (green) in Figure 8b after the failure.

Figure 8c shows that the total time to complete reception is longer than in Scenario 1. This is a direct result of the reduced goodput, confirming the observations in Figure 8a.

Overall, Scenario 2 highlights the importance of careful link selection when configuring FRER. Since an MLD STA can associate with only one AP under the current WiFi standard, it is not feasible to configure node-disjoint paths for redundancy. However, this limitation may be addressed in WiFi 8, which introduces support for multi-AP coordination [29]. To mitigate the effects of asymmetry, one may prefer multiple links with similar capacity and speed such as channels within the same frequency band, comparable frequency bands, or modulation schemes offering equivalent throughput characteristics.

Scenario 3. Figure 9 presents the analysis of Scenario 3, where i_3 of the AP (i.e., the slower interface of the AP). This scenario introduces two key differences compared to the previous ones: First, the failure of i_3 causes the AP to

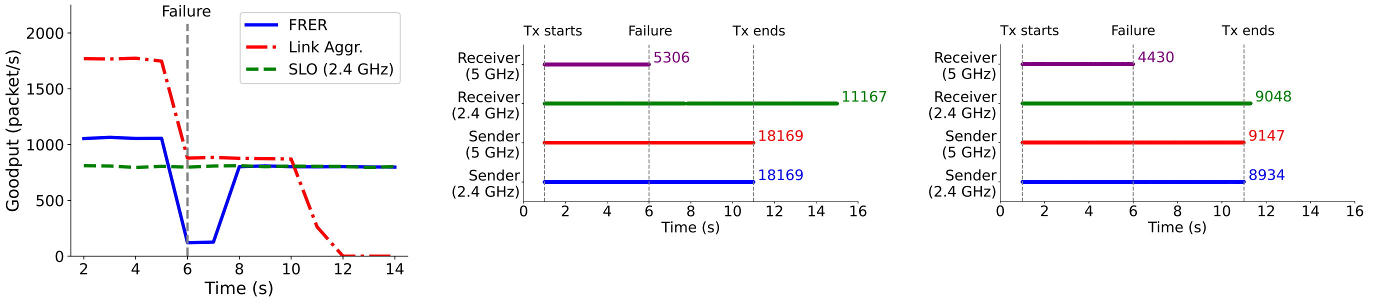


(a) Goodput over time.

(b) Tx and Rx timeline in FRER mode.

(c) Tx and Rx timeline in link aggregation mode.

Figure 9: Scenario 3 – Failure of the AP's interface at 2.4 GHz (i_3 at AP).



(a) Goodput over time.

(b) Tx and Rx timeline in FRER mode.

(c) Tx and Rx timeline in link aggregation mode.

Figure 10: Scenario 4 – Failure of the AP's interface at 5 GHz (i_4 at AP).

lose connectivity over two links, l_1 and l_3 (see Figure 5), effectively resulting in a multi-link failure. Second, the AP becomes a bottleneck, as the sender continues to utilize multiple links, while the AP is left with only one operational interface. The impact of these differences is illustrated in Figure 9a. In the link aggregation mode, the sender keeps transmitting packets through its available interfaces even after the failure at the AP. However, frames sent over the 2.4 GHz interface (i_1) are either queued in the L-MAC or lost, as they are no longer received by the AP. This leads to a notable drop in goodput. By contrast, in the FRER mode, the AP continues to receive one replica of each original frame via its remaining 5 GHz interface and forwards it to the receiver. This maintains seamless redundancy and results in better performance than the link aggregation mode after the failure.

Figure 9b and 9c confirm this behavior. The sender continues transmitting over both interfaces, while the AP only receives and forwards frames over the 5 GHz interface (purple). One possible mitigation for such interface failure scenarios is to enable the sender's U-MAC to process association beacons from the AP. If beacons are not received (indicating a disconnection), the sender can stop forwarding packets to the affected interface. However, similar issues may still occur under other conditions, such as severe channel noise or reception-side failures at the AP interface (while transmission remains functional).

Scenario 4. Figure 10 presents the analysis of Scenario 4, where i_4 of the AP fails. The results reflect a combination of two cases observed in Scenarios 2 and 3. First, in the FRER mode, we observe a drop in goodput due to the use of asymmetrical links, as discussed in Scenario 2. Immediately after the failure, the AP receives duplicate frames via the 2.4 GHz interface (i_3), which it had already forwarded earlier through the now-failed 5 GHz link. These redundant frames are eliminated between $t = [6s, 7s]$, after which newly arriving frames are forwarded via i_3. Second, in the link aggregation mode, the goodput experiences a sharper decline. This is because the sender's U-MAC continues to transmit to the now-nonfunctional interface of AP, as described in Scenario 3.

Scenario	**Packet Delivery Ratio (%)**		
	FRER	Link Aggr.	SLO
Scenario 1	100.0	100.0	32.3
Scenario 2	100.0	100.0	N/A
Scenario 3	100.0	74.1	31.4
Scenario 4	100.0	74.5	N/A

Table 5: Packet delivery ratio across different scenarios.

Finally, Table 5 summarizes the packet delivery ratio of SLO and MLO in both FRER and link aggregation modes across the failure scenarios. Note that these values are influenced by the timing of the respective failures. For example, an earlier failure leads to greater packet loss over the scenario duration. As shown in the results, FRER consistently achieves a 100% delivery ratio, even when

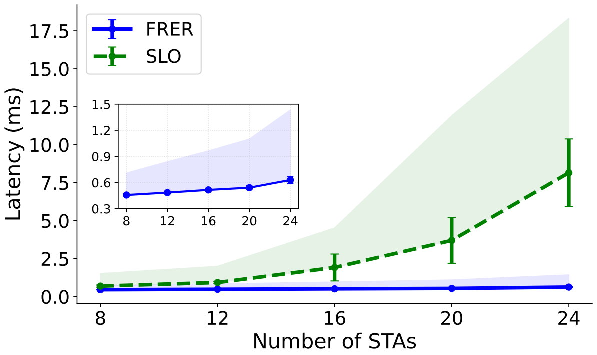


Figure 11: Scenario 5 – Latency in mixed MLO and SLO congestion scenarios.

goodput fluctuates under certain conditions (see Scenarios 2 and 4). In contrast, the link aggregation mode suffers more than 25% packet loss, as the sender cannot detect failures at the AP but continues transmitting over the remaining operational link. SLO is susceptible to failures, resulting in a complete halt in packet delivery once a failure occurs.

6.5. Congestion Scenarios

We evaluate the impact of network congestion by increasing the number of stationary STAs up to 24. Half of these nodes are designated as senders, and the other half as receivers of UDP traffic. Among them, only one sender–receiver pair is FRER-enabled; the remaining STAs operate in SLO mode, each transmitting over a randomly selected link, either 2.4 GHz or 5 GHz. As the number of STAs increases, channel contention is expected to intensify, leading to higher latency.

Scenario 5. Figure 11 and 12 present the results of Scenario 5. Figure 11 compares the mean (lines) and 95th percentile (areas) latency between FRER (blue, solid) and SLO (green, dashed) modes. For FRER, the measurements reflect end-to-end latency between the FRER-enabled STA pair, where only the earlier-arriving replica is considered for each packet. For SLO, latency is averaged across all packet exchanges between SLO STAs in a given simulation run. We keep the traffic load for each STA fixed when increasing their number. Each experiment is repeated 10 times.

The results in Figure 11 clearly demonstrate that latency in the FRER mode remains nearly constant at around 1 ms, whereas SLO STAs experience rapidly increasing latency with high variance, reaching up to 17.5 ms. As shown in the zoomed-in view, the presence of redundant packets in FRER effectively bounds the latency, even under increased contention. However, as previously discussed in Scenarios 1–4, FRER introduces additional resource overhead due to packet replication. This not only wastes channel capacity but can also exacerbate congestion for coexisting SLO STAs, degrading their performance. Therefore, FRER mode should be reserved for high-criticality traffic, where strict reliability and bounded latency are essential, aligning with the core objectives of TSN.

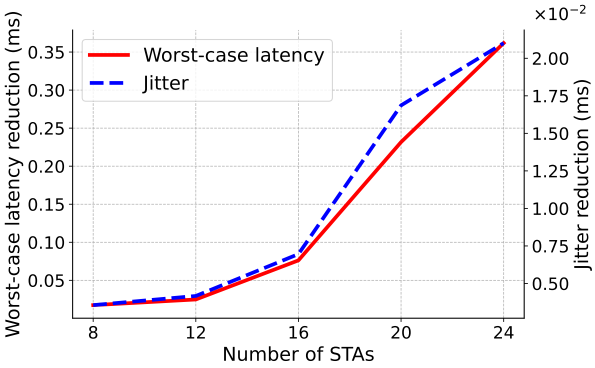


Figure 12: Scenario 5 – Normalized worst-case latency and jitter reduction in congestion scenarios. The normalization represents the improvement achieved per additional 1% of redundant packets transmitted via FRER.

Figure 12 further illustrates the normalized latency reduction (NLR) in terms of worst-case latency (left axis, red solid line) and jitter improvements (right axis, blue dashed line) provided by redundancy compared to SLO. The worst case corresponds to the 95th percentile latency presented in Figure 11. We consider these two metrics as representative of the reliability improvements achieved by FRER regarding bounded and predictable latency. As shown in the figure, both worst-case latency and jitter reductions become more significant with an increasing number of STAs, i.e., under higher congestion. Specifically, we observe up to a 0.35 ms improvement in worst-case latency per additional 1% of redundant packets in FRER. The same trend is observed for jitter improvements, indicating more stable and reliable communication delays.

In Scenario 5, we aim for a preliminary investigation on the benefits of FRER for a low and predictable latency, and for demonstrating this in our integrated FRER-MLO framework. This can be extended with further experimentation on more detailed cases such as: (i) increasing number of FRER-enabled STAs (instead of only one pair), a more controlled distribution of SLO STAs to different frequency bands (instead of random assignment), (iii) changing channel conditions, and (iv) indoor scenarios. Our open-source framework enables simulating all these cases, and we also consider such experiments as our future work.

6.6. Mobility Scenarios

Finally, we evaluate the latency improvements in FRER mode compared to legacy SLO STAs under mobility scenarios. In the experiments, 16 STAs move according to random waypoint mobility with a speed up to 9 m/s in a larger area of 150 m × 150 m. While slow speeds (e.g., 1 m/s) represents human walking, faster ones can resemble with mobile robots and automated guided vehicle mobility in large industrial facilities with WiFi networking.

Scenario 6. Figure 13 and 14 present the results of Scenario 6. In contrast to the congestion results observed in

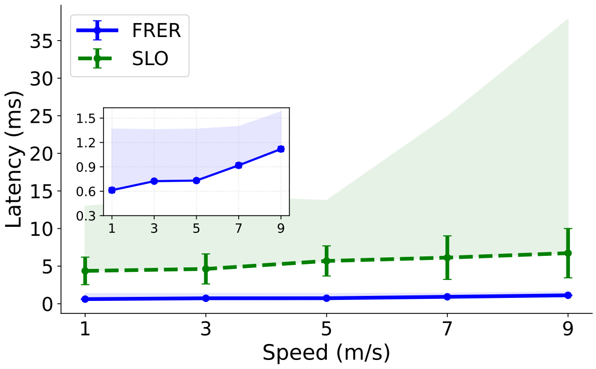


Figure 13: Scenario 6 – Latency in mobile scenarios.

Scenario 5, we observe a more steady increase in average latency for SLO STAs in Figure 13. However, the worst-case latency at the 95th percentile peaks more sharply, as increasing mobility leads to dynamically changing channel occupancy and, consequently, higher contention. Furthermore, during the experiments, SLO STAs operating at 5 GHz were more susceptible to increased distances, often requiring retransmissions when they moved beyond the effective transmission range. In comparison, the benefits of FRER are clearly observed. It provides multiple opportunities to transmit the same packets over two links and can maintain a stable connection by falling back to the 2.4 GHz band when the 5 GHz link becomes unstable.

Figure 14 shows slightly different NLR results compared to the congestion scenarios. For both worst-case latency and jitter, the improvements are more pronounced, reaching up to 0.8 ms and 3.7×10^{-2} ms per additional 1% of redundant traffic, respectively. For instance, with 50% redundant traffic as in our evaluation scenario (i.e., twofold duplication), the jitter reduction is approximately 1.85 ms, which is considerable for time-sensitive systems with strict latency requirements of 5–10 ms.[7]

7. Open Issues and Challenges

In this section, we outline some key open implementation and configuration challenges.

Configuration Complexity. Currently, FRER requires detailed configuration across multiple modules on every FRER-enabled node. The native FRER implementation in OMNeT++ includes the `StreamRedundancyConfigurator` to automate this process. However, this module only supports TSN devices, as it enforces VLAN- and PCP-based configurations. Enabling VLAN and PCP tagging, alongside using this configurator in our framework, not only facilitates scalable network simulation but also lays the groundwork for simulating hybrid TSN and WiFi networks with end-to-end redundancy.

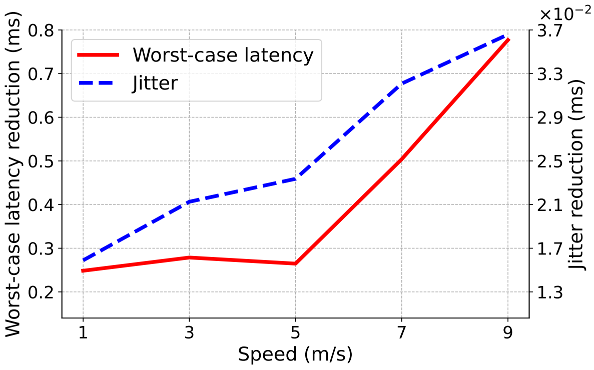


Figure 14: Scenario 6 – Normalized worst-case latency and jitter reduction in mobile scenarios. The normalization represents the improvement achieved per additional 1% of redundant packets transmitted via FRER.

VRA Window Size. Another critical configuration parameter for FRER is the VRA window size (see Section 5.1.2). Ethernet-based TSN setups generally assume two or more disjoint paths with similar latency. Under such conditions, a small VRA window is sufficient, as duplicate frames arrive at the SRF of a TSN bridge or listener closely spaced and in order. In contrast, wireless networks experience retransmissions and asymmetrical links, causing sequence numbers of simultaneously arriving packets to diverge. Consequently, older frames may be forwarded without elimination because they fall outside the current VRA window. In our experiments, we set the window size to 1000 and observed that values below 500 did not function correctly. Hence, this parameter should be carefully tuned based on network characteristics. For further details, readers may refer to ongoing standardization efforts [30] and alternative methods [31].

Individual Recovery Function. For stream recovery, we implement only an SRF above the U-MAC layer, rather than deploying an IRF per link at the L-MAC. In TSN bridges, the IRF eliminates duplicate packets earlier at the receiving port and reduces processing load on the bridging module, which is especially useful against stuck senders. A more realistic FRER integration could incorporate this approach, albeit requiring significant modifications to the current L-MAC design in OMNeT++.

8. Conclusion

The new multi-link operation (MLO) feature of WiFi 7 enables the simultaneous use of multiple interfaces across the 2.4, 5 and 6 GHz bands. This can significantly improve spectrum efficiency and increase reliability under challenging wireless link conditions. A promising approach

[7] Please note that we discuss the implications of NLR specifically in the context of FRER and MLO. For other redundancy or retransmission mechanisms, such a direct correlation may not necessarily apply.

to enhance reliability is to introduce redundancy by leveraging spatial and frequency diversity across multiple links. A prominent TSN protocol, IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER), provides mechanisms for flexible packet replication and elimination in wired Ethernet networks, which could similarly be utilized to achieve redundancy in WiFi.

Accordingly, in this paper, we present an open-source framework that integrates FRER with MLO in the network simulator OMNeT++, extending our previous work in [7]. We address several implementation challenges and examine the implications of adapting FRER-based redundancy to wireless networks. Through several scenarios of interface failures, link congestion, and mobility scenarios, we evaluate the effects of asymmetrical wireless links and non-disjoint paths configured for FRER, and propose potential solutions. We also demonstrate the benefits of FRER in terms of latency and packet delivery ratio, compared to legacy SLO and another MLO mode, link aggregation. Our results show that FRER can enhance packet delivery ratio and ensure bounded latency, albeit at the cost of reduced channel efficiency due to redundancy.

For future work, we aim to improve channel efficiency by employing deferral techniques in FRER, which avoid redundant transmissions under favorable channel conditions when at least one replica has been successfully delivered. Furthermore, we plan to investigate the integration of wireless FRER into hybrid TSN and WiFi networks, enabling holistic end-to-end redundancy across both wireless and wired domains.

Acknowledgment

This work was supported by the OWTSN project as a part of TARGET-X and 6G Joint Undertaking for Intelligent Networks and Services (6G SNS JU) programs under Horizon Europe grant number 101096614 as well as by the project ML4WIFI funded by the German Research Foundation (DFG) under grant number DR 639/28-1.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used AI tools only for proofreading and improving the language. After their use, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

References

[1] Evgeny Khorov, Ilya Levitsky, and Ian F. Akyildiz. Current Status and Directions of IEEE 802.11be, the Future Wi-Fi 7. *IEEE Access*, 8:88664–88688, May 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2993448.

[2] Doğanalp Ergenç and Mathias Fischer. On the Reliability of IEEE 802.1CB FRER. In *40th IEEE International Conference on Computer Communications (INFOCOM 2021)*, pages 1–10, Virtual Conference, May 2021. IEEE. doi: 10.1109/infocom42981.2021.9488750.

[3] Doğanalp Ergenç and Mathias Fischer. Implementation and Orchestration of IEEE 802.1CB FRER in OMNeT++. In *IEEE International Conference on Communications (ICC 2021), Workshop on Time-sensitive and Deterministic Networking (TsDn 2021)*, pages 1–6, Virtual Conference, June 2021. IEEE. doi: 10.1109/ICCWorkshops50388.2021.9473722.

[4] Kota Nikhileswar, Krishnanand Prabhu, Dave Cavalcanti, and Alon Regev. Time-Sensitive Networking Over 5G for Industrial Control Systems. In *27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, Stuttgart, Germany, September 2022. IEEE. ISBN 978-1-66549-996-5. doi: 10.1109/ETFA52439.2022.9921680.

[5] Dave Cavalcanti, Carlos Cordeiro, Malcolm Smith, and Alon Regev. WiFi TSN: Enabling Deterministic Wireless Connectivity over 802.11. *IEEE Communications Standards Magazine*, 6 (4):22–29, December 2022. ISSN 2471-2825. doi: 10.1109/MCOMSTD.0002.2200039.

[6] Juan Fang, Susruth Sudhakaran, Dave Cavalcanti, Carlos Cordeiro, and Cheng Chen. Wireless TSN with Multi-Radio Wi-Fi. In *IEEE Conference on Standards for Communications and Networking (CSCN 2021)*, pages 105–110, Thessaloníki, Greece, December 2021. IEEE. doi: 10.1109/CSCN53733.2021.9686180.

[7] Doğanalp Ergenç and Falko Dressler. An Open Source Implementation of Wi-Fi 7 Multi-Link Operation in OMNeT++. In *20th IEEE/IFIP Wireless On-demand Network systems and Services Conference (WONS 2025)*, pages 131–134, Hintertux, Austria, January 2025. IEEE. ISBN 978-3-903176-71-3.

[8] IEEE. IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability. Std 802.1cb-2017, IEEE, October 2017.

[9] William Stallings. IEEE 8O2.11: Wireless LANs from a to n. *IT Professional*, 6(5):32–37, 2004. ISSN 1941-045X. doi: 10.1109/MITP.2004.62.

[10] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. Std 802.11e-2005, Institute of Electrical and Electronics Engineers (IEEE), 11 2005.

[11] Stefano Avallone, Pasquale Imputato, Getachew Redieteab, Chittabrata Ghosh, and Sumit Roy. Will OFDMA Improve the Performance of 802.11 Wifi Networks? *IEEE Wireless Communications*, 28(3):100–107, 6 2021. ISSN 1536-1284. doi: 10.1109/MWC.001.2000332.

[12] Abinash Sinha, Reiner Stuhlfauth, and Fernando Schmitt. In-device Coexistence Interference. Online article, Microwave Journal, 4 2018. URL https://www.microwavejournal.com/articles/26609-in-device-coexistence-interference.

[13] Toni Adame, Marc Carrascosa-Zamacois, and Boris Bellalta. Time-Sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7. *MDPI Sensors*, 21(15):4954, July 2021. ISSN 1424-8220. doi: 10.3390/s21154954.

[14] 802.11ak-2018 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Enhancements for Transit Links Within Bridged Networks. Std 802.11ak-2018, IEEE, June 2018.

[15] A. de la Oliva, X. Wang, R. Yang, and R. Gazda. Improving WLAN reliability. Submission ieee 802.11/11-19-1223-00-00be, IEEE, March 2019. URL https://mentor.ieee.org/802.11/dcn/19/11-19-1223-00-00be-improving-wlan-reliability-joint-tsn-11be-session.pdf.

[16] Lorenzo Galati-Giordano, Giovanni Geraci, Marc Carrascosa-Zamacois, and Boris Bellalta. What will Wi-Fi 8 Be? A Primer on IEEE 802.11bn Ultra High Reliability. *IEEE Communications Magazine*, 62(8):126–132, 8 2024. ISSN 0163-6804. doi: 10.1109/MCOM.001.2300728.

[17] T. Montgomery, Susruth Sudhakaran, Mohamed Hany Kashef, Jing Geng, Dave Cavalcanti, and Richard Candell. IEEE 802.1CB Frame Elimination in a Wireless HSR Architecture. In *50th Annual Conference of the IEEE Industrial Electronics Society (IECON 2024)*, pages 1–7, Chicago, IL, 11 2024. Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/IECON55916.2024.10905686.

[18] Ammad Ali Syed, Serkan Ayaz, Tim Leinmüller, and Madhu Chandra. Fault-Tolerant Dynamic Scheduling and Routing for TSN based In-vehicle Networks. In *13th IEEE Vehicular Networking Conference (VNC 2021)*, pages 72–75, Virtual Conference, November 2021. IEEE. ISBN 978-1-66544-450-7. doi: 10.1109/VNC52810.2021.9644662.

[19] Guizhen Li, Shuo Wang, Yudong Huang, Tao Huang, Yuanhao Cui, and Zehui Xiong. Optimizing Fault-Tolerant Time-Aware Flow Scheduling in TSN-5G Networks. *IEEE Transactions on Mobile Computing*, 24(4):3441–3455, 2025. ISSN 1536-1233. doi: 10.1109/TMC.2024.3510604.

[20] Adnan Aijaz. 5G Replicates TSN: Extending IEEE 802.1CB Capabilities to Integrated 5G/TSN Systems. In *IEEE Conference on Standards for Communications and Networking (CSCN 2024)*, pages 108–112, Belgrade, Serbia, November 2024. doi: 10.1109/CSCN63874.2024.10849690.

[21] Pierre E. Kehl, Junaid Ansari, Mikael Lovrin, Praveen Mohanram, Chi-Chuan Liu, Jun-Lin Yeh, and Robert H. Schmitt. 5G-TSN Integrated Prototype for Reliable Industrial Communication Using Frame Replication and Elimination for Reliability. *Electronics*, 14(4), 2025. ISSN 2079-9292. doi: 10.3390/electronics14040758.

[22] Marie-Theres Suer, Christoph Thein, Hugues Tchouankem, and Lars Wolf. Adaptive Multi-Connectivity Scheduling for Reliable Low-Latency Communication in 802.11be. In *IEEE Wireless Communications and Networking Conference (WCNC 2022)*, pages 102–107, Austin, TX, April 2022. IEEE. ISBN 978-1-66544-267-1. doi: 10.1109/WCNC51071.2022.9771939.

[23] A. Jauh, Y. Hsu, H. Yu, L. Wang, Zuo X., and K. Meng. Conditional Packet Duplication in Multi Link System. Submission ieee 802.11-19/1101r1, IEEE, July 2019. URL `https://mentor.ieee.org/802.11/dcn/19/11-19-1101-01-00be-conditional-packet-duplication-in-multiple-link-system.pptx`.

[24] Gianluca Cena, Stefano Scanzio, Dave Cavalcanti, and Valerio Frascolla. Seamless Redundancy for High Reliability Wi-Fi. In *19th IEEE International Conference on Factory Communication Systems (WFCS 2023)*, Pavia, Italy, April 2023. IEEE. doi: 10.1109/WFCS57264.2023.10144228.

[25] Gianluca Cena, Stefano Scanzio, and Adriano Valenzano. Seamless Link-Level Redundancy to Improve Reliability of Industrial Wi-Fi Networks. *IEEE Transactions on Industrial Informatics*, 12(2):608–620, 2016. ISSN 1551-3203. doi: 10.1109/TII.2016.2522768.

[26] Gianluca Cena, Stefano Scanzio, and Adriano Valenzano. Experimental Evaluation of Seamless Redundancy Applied to Industrial Wi-Fi Networks. *IEEE Transactions on Industrial Informatics*, 13(2):856–865, April 2017. ISSN 1551-3203. doi: 10.1109/TII.2016.2641469.

[27] Azucena García-Palacios, Stefano Scanzio, and Adriano Valenzano. Improving Effectiveness of Seamless Redundancy in Real Industrial Wi-Fi Networks. *IEEE Transactions on Industrial Informatics*, 14(5):2095–2107, May 2018. ISSN 1551-3203. doi: 10.1109/TII.2017.2759788.

[28] Roger B. Marks and Norm Finn. Clarifying EPD and LPD. Document id: maint-marks-finn-hlpde-1119-copyright, IEEE, November 2019. URL `https://www.ieee802.org/1/files/public/docs2019/maint-Marks-Finn-epd-lpd-1119-copyright.pdf`.

[29] Lorenzo Galati-Giordano, Giovanni Geraci, Marc Carrascosa, and Boris Bellalta. What will Wi-Fi 8 Be? A Primer on IEEE 802.11bn Ultra High Reliability. *IEEE Communications Magazine*, 62(8):126–132, August 2024. ISSN 0163-6804. doi: 10.1109/mcom.001.2300728.

[30] P802.1CBec – Guidance for Sequence Recovery Function Parameter Configuration. Std P8021CBec, IEEE, April 2025. URL `https://1.ieee802.org/tsn/802-1cbec/`.

[31] Lisa Maile, Dominik Voitlein, Kai-Steffen Jens Hielscher, and Reinhard German. Ensuring Reliable and Predictable Behavior of IEEE 802.1CB Frame Replication and Elimination. In *IEEE International Conference on Communications (ICC 2022)*, pages 2706–2712, Seoul, South Korea, May 2022. IEEE. doi: 10.1109/ICC45855.2022.9838905.