L25MCP: Flexible Multi-connectivity in WiFi Networks

David Niederprüm
TU Braunschweig
Braunschweig, Germany
niederpruem@ibr.cs.tu-bs.de

Christian Dietrich
TU Braunschweig
Braunschweig, Germany
dietrich@ibr.cs.tu-bs.de

Lars Wolf
TU Braunschweig
Braunschweig, Germany
wolf@ibr.cs.tu-bs.de

***Abstract*—Deploying WiFi in production-critical manufacturing systems is challenging due to uncontrolled interference and unpredictable latencies. Existing redundancy solutions either require separate broadcast domains or lack scheduling flexibility due to fixed hardware implementations (WiFi 7 MLO).**

This paper presents the Layer 2.5 Multi-connectivity Protocol (L25MCP), an application-transparent protocol that enables fine-grained, per-packet combinations of load balancing, duplication, and splitting within a single Layer 2 broadcast domain. L25MCP utilizes a compatible ARP extension (L25ARP) to map a single IP address to multiple physical MAC addresses.

Real-world evaluation in a WiFi 7 testbed shows that application-specific MC schemes implemented with L25MCP can reduce the residual probability of consecutive deadline misses by a factor of 212 compared with off-the-shelf MLO schemes. We also show that selective packet duplication reduces application latency by preventing channel congestion.

***Index Terms*—Multi connectivity, wireless reliability, real-world implementation**

I. Introduction

Smart production systems increasingly incorporate mobile devices [1], such as Automated Guided Vehicles (AGVs), whose operation depends on wireless communication. While wire-attached devices operate on a rather stable transmission medium, communication disruptions are inherently more severe for mobile nodes and can result in costly production interruptions. Consequently, a high level of communication reliability is a prerequisite for the deployment of wireless technologies in production-critical industrial systems.

WiFi is an attractive wireless communication technology increasingly used in industrial settings. Its widespread adoption in consumer and enterprise markets has produced low-cost hardware, a mature device ecosystem, and easy integration with standard Ethernet/IP infrastructures. However, WiFi was not designed with reliability in mind: operating in unlicensed ISM bands makes it vulnerable to uncontrolled interference, and its shared-medium and contention-based channel access can cause collisions, retransmissions, and unpredictable latency. Despite its inability to provide hard real-time guarantees, WiFi is often used for soft real-time and best-effort industrial applications [2].

Several techniques that use multiple WiFi links concurrently have been proposed to transparently improve reliability for applications (see Tab. I). At the protocol level, the Parallel Redundancy Protocol (PRP) [3], which was designed to provide redundancy for wired Ethernet networks, and FRER[1], which targets the Time Sensitive Networking (TSN) domain, were adapted for use with WiFi [5], [6]. Both protocols duplicate all traffic and send it to separate broadcast domains. While this provides spatial diversity, it requires the operator to set up redundant network infrastructures (see Fig. 1). Also, in congested WiFi networks, redundancy schemes that duplicate traffic unconditionally add extra load and thus further degrade performance.

TABLE I: Characteristics of L25MCP in comparison to PRP and MLO

	L25MCP	**PRP**	**MLO**
Application transparency	✓	✓	✓
Compatible with existing network infrastructure	✓	✗	(✓)
Spatial diversity	✓	✓	✗
Flexible Multi-Connectivity schemes (also *adaptive*)	✓	✗	(✓)
Per-packet Multi-Connectivity (MC)	✓	✗	(✓)

At the network-infrastructure level, the IEEE 802.11 standard introduces Multi Link Operation (MLO) with WiFi 7 [7]. In contrast to PRP and FRER, MLO does not duplicate traffic but balances it across multiple links to increase throughput and reduce latency. To benefit from MLO, an operator must upgrade the entire network to WiFi 7, which requires upfront costs and complicates integration of existing production machines. In principle, WiFi 7 allows per-packet steering by assigning a set of links to each of the eight user-definable WiFi traffic classes [7], [8]. However, in practice, product vendors may choose whether to expose this mapping as a configurable option or to use a proprietary scheduler. Unlike PRP/FRER, MLO offers load balancing without spatial diversity: At one point in time, MLO devices are connected only to a single AP, whereas with PRP, a station connects to two different APs (see Fig. 1).

Besides these practical implementations, prior academic research has explored flexible methods to leverage multiple links [9], [10], [11], [12], [13]. Suer et al. [14] provide a taxonomy for designing *Multi-Connectivity* (MC) schemes that improve different non-functional properties (reliability,

[1]for FRER: IEEE 802.1CB-2017 Frame Replication and Elimination for Reliability [4]

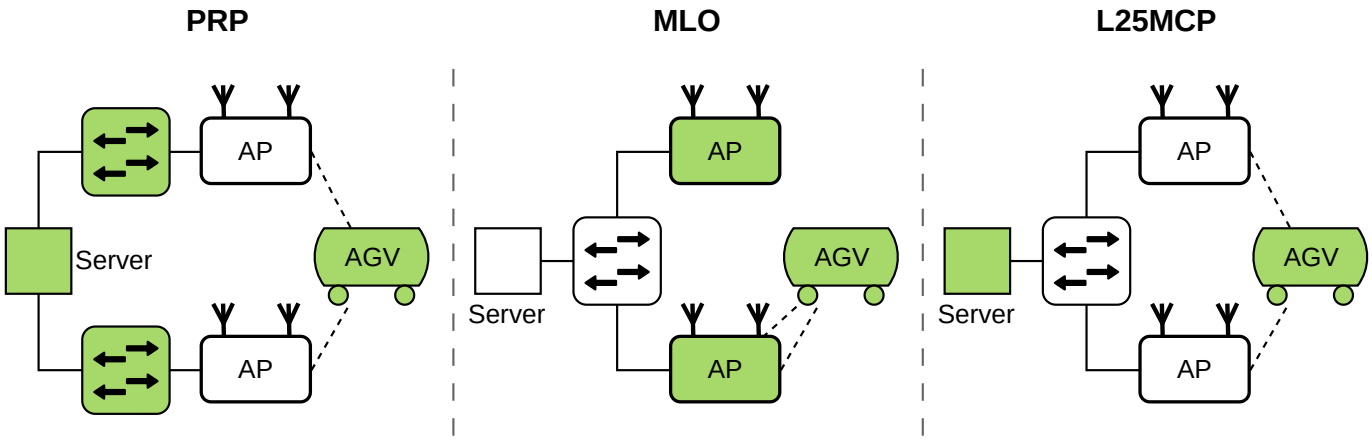


Fig. 1: Comparison of L25MCP's topology with Multi Link Operation (MLO) and PRP. MLO is limited by connecting to only one Access Point (AP) at a time. PRP requires two separate L2 broadcast domains. Green denotes the devices that need to be altered for the respective protocol.

latency, throughput, ...). They identify three categories of schemes that can be applied individually or in combination: Load Balancing (LB), Packet Duplication (PD), and Packet Splitting (PS). Building on this taxonomy, they explored adaptive techniques that reconfigure the redundancy scheme based on channel metrics [12], [13]. Still, the effectiveness of such adaptive schemes in real-world testbeds remains understudied.

The missing element for exploring flexible MC schemes is a protocol and a practical implementation that can be deployed and evaluated on a real-world testbed. None of PRP, FRER, or MLO is flexible enough to realize mixed MC schemes or to switch dynamically between schemes. For example, in a high-contention scenario, we would like to disable duplication for low-priority traffic and switch to load-balancing. Also, for a real-world testbed, nodes have to exchange MC information and inform each other about their capabilities.

To this end, we present the *Layer 2.5 Multi-connectivity Protocol* (see L25MCP in Tab. I). Similar to PRP and MLO, the L25MCP provides an application-transparent solution for MC. In contrast to PRP, L25MCP does not require a fully redundant network infrastructure as it operates in a single ISO/OSI layer 2 (L2) broadcast domain (see Fig. 1). L25MCP devices can connect to different APs at the same time to get spatial diversity and L25MCP is not limited to a single MC scheme. It can be used to realize LB, PD, and PS and combinations thereof. Also, the L25MCP nodes find each other with an extension of L2 address resolution.

In particular, we claim the following contributions:

1) The L25MCP as a generic protocol for MC within a networking broadcast domain.
2) An application-transparent L25MCP implementation in Linux user-space that eases exploration of novel MC schemes.
3) A real-world evaluation of fine-grained MC schemes for two example scenarios with L25MCP.

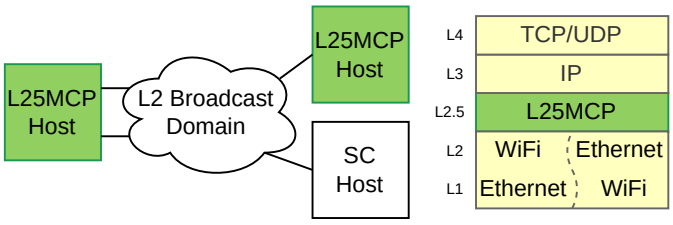


Fig. 2: Overview of L25MCP; L25MCP-capable hosts maintain one or more connections in the same L2 network. The protocol does not interfere with single-connectivity hosts. It maps multiple L2 devices to one IP layer.

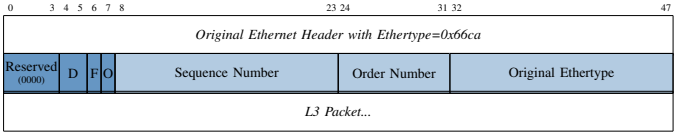
II. L25MCP

L25MCP is a protocol for MC in Local Area Networks (LANs) with a single L2 broadcast domain. It sits between the WiFi/Ethernet and the IP layer (see Fig. 2 right) and is designed to be compatible with existing network infrastructure. This includes the ability for an L25MCP-capable host to participate with multiple L2 links in a single broadcast domain.

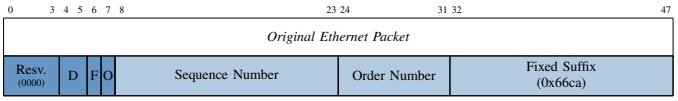
L25MCP is not limited to a single MC scheme; it supports the three basic MC operations (i.e., LB, PD, PS) on the L2 payload, individually or in combination. Towards the IP protocols on the ISO/OSI layer 3 (L3), L25MCP appears as a normal L2 protocol. To benefit from L25MCP, we only require that two or more L2 paths exist between two hosts. As exemplified in Fig. 1, it is sufficient for the wireless host to have multiple network interfaces while its peer can be connected via a single Ethernet link. Also, L25MCP is backwards compatible and L25MCP nodes can communicate with regular single-connectivity hosts (see Fig. 2).

A. L25ARP: L2 address resolution for L25MCP

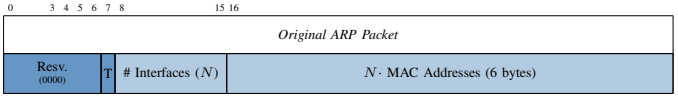
In the network, L25MCP hosts must recognize each other to upgrade their communication to MC. Also, we have to exchange information, like MAC addresses, about the MC



(a) L25MCP protocol (header mode)



(b) L25MCP protocol (trailer mode)



(c) L25ARP extension

Fig. 3: L25MCP Protocol Diagrams (blue for added fields, flags of the type field in darker blue)

setup. For this, we present L25ARP, a compatible extension to the ARP protocol.

With multiple links in one L2 network, we face the problem that existing protocols for address resolution (Address Resolution Protocol (ARP) and Neighbour Discovery Protocol (NDP)) resolve a *single* IP address to a single MAC address. Other redundancy protocols, like PRP, avoid this problem by requiring us to set up separate L2 broadcast domains, which allows them to use the same MAC address on both interfaces. While this is compatible with standard address resolution, it complicates the network infrastructure. With L25ARP, one L3 maps to *multiple* MAC addresses for a host with multiple interfaces.

We adapt ARP by extending its messages as shown in Fig. 3c. Because IPv6's NDP works similarly, the same solution applies to NDP. Whenever a L25MCP host issues or responds to an ARP query, it sends the regular ARP message carrying its IPv4 address and appends the MAC addresses of all its network interfaces ($\geq$1) that reside in the same L2 domain. Additionally, we encapsulate the L25ARP packet as an L25MCP packet (see Sec. II-B). Since ARP messages are small and infrequent, L25MCP duplicates them across all connected interfaces (PD). Hence, multi-link hosts receive their own broadcast messages; L25MCP detects these duplicates based on the sender address and drops them. Single-linked L25MCP hosts also send L25ARP packets but include only their single MAC in the extended part to identify themselves as L25MCP-capable. After an L25ARP exchange, two L25MCP hosts know (a) that the other side is an L25MCP node and (b) all MAC addresses of the peer.

B. L25MCP Protocol Data

If we want to apply the different MC schemes in a fine-grained manner, we have to add additional information in each packet to allow the receiver to reverse the applied MC transformation. For PD, the receiver must be able to drop duplicate packets if he receives more than one copy. For PS, the different packet segments have to be reassembled into a single packet. For LB, no additional information is required in principle but L3 could benefit from in-order delivery.

L25MCP can either use the header or trailer mode to transport this MC information (see Fig. 3a and Fig. 3b). In the header mode, we insert a 6-byte header between L2 and L3 (similar to the Redundancy Tag (R-Tag) of FRER). And in trailer mode, we append the information after the L3 packet ends (as done by PRP). Both methods have benefits and shortcomings: While the header systematically encloses the L3 payload and avoids every chance to confuse L2.5 packets with normal L2 packets, it breaks compatibility with WiFi Quality of Service (QoS). This is because WiFi APs are unaware of L25MCP and therefore are unable to read the Differentiated Services Code Point (DSCP) from the IP header, which determines the WiFi QoS traffic class. The trailer mode avoids this drawback but provokes an uncommon packet where the IP payload does not align with the frame length. In our experiments, we experienced that some APs cut off the trailer when bridging between WiFi and Ethernet.

Hence, L25MCP supports both modes and hosts communicate via the **T** flag in the L25ARP packet whether they use the trailer mode and whether their peer should expect a trailer. In header mode, we replace the original IP header with the L25MCP header and save the original Ethertype. In trailer mode, we set 16-bit fixed *suffix* to perform heuristic in-band identification of L25MCP packets. However, as both ends of the communication learned in the L25ARP exchange if they should expect L25MCP packets with a trailer, collisions can be avoided.

We cover the basic MC schemes and their combination by adding three fields to each packet: *type* (1 byte), *sequence number* (2 bytes), and *order number* (1 byte). The sequence number is a host global counter that is incremented on each packet sent. The order number is per packet. The *type* describes the applied MC transformation and how to reverse it, the upper 4 bits are reserved for future use and currently default to zero. The lower four bits have the following orthogonal functions:

D field This two-bit field indicates the number of *additional* duplicates that were sent. The receiver should perform deduplication and the duplicates are identified by the sequence number.

F flag If set, the packet was split and this packet is only a fragment. The receiver has to reassemble the packet, for which he has to use the order number to arrange the fragments. The last fragment is identified by having the most-significant bit in the order number set. So, a packet can be split into 128 fragments.

O flag If set, the receiver should make an effort to deliver packets in-order to L3 according to the sequence number. This can improve performance of higher layers that expect their packets to usually arrive in order.

With these four orthogonal flags, we can express different kinds of MC operations at the protocol level. For example, when we receive a packet from a host with three interfaces

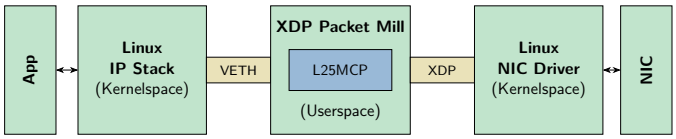


Fig. 4: Implementation of L25MCP in Linux using the XDP Packet Mill (XPM) [15]. IP packets from the network stack travel through a Virtual Ethernet (VETH) tunnel into the L25MCP layer, which applies an MC scheme for transmission with multiple physical network interfaces.

with D=1,O=1, we know that the other side created two duplicates (and probably performs load balancing) and that we should reorder the received packets before we give it to L3.

III. L25MCP Prototype for Linux

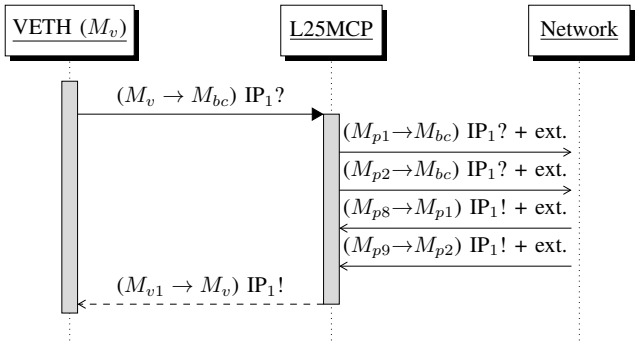
We implement L25MCP for Linux; the prototype transparently intercepts application traffic and encapsulates it in our protocol. We build on the XDP Packet Mill (XPM) [15], a framework for transparent packet interception that uses VETH tunnels and eXpress Data Path (XDP), both novel Linux interfaces for high-performance networking. Using this prototype, we evaluate L25MCP on a real-world wireless testbed in Sec. IV.

As shown in Fig. 4, XPM connects to the application side via a VETH tunnel. This lets the application use the regular `socket(2)` interface while the Linux IP stack processes the data as usual, placing the Ethernet frame into the VETH interface. On the hardware side, XPM connects to the kernel's NIC driver via the XDP interface, providing access to the Network Interface Card (NIC) without traversing the Linux IP stack. XPM runs as a user-space process, intercepting L2 packets from both sides and it offers operations to drop, slice, or duplicate individual packets.

For L25MCP, we translate Single Connectivity (SC) traffic from the VETH tunnel into MC traffic for transmission on the physical interfaces, and vice versa. With XPM's packet operations, we implement the L25MCP protocol and variations of LB, PD, and PS.

Special consideration has to be taken into account for the L25ARP protocol. To keep the implementation as simple as possible, we rely on Linux' ARP cache and extend it by an L25ARP cache that is stored in the XPM. When the application wants to communicate to an IP address (see Fig. 5) for which Linux' ARP cache has no entry, the kernel will send an ARP request over the VETH device. L25MCP receives this request and sends out N L25ARP requests, which contain the MACs of all physical interfaces, to every connected physical device (PD) and waits for answers.

At the receiving end of a L25ARP packet (request or response), we store the received information in the L25ARP cache and interface with the kernel's ARP cache. Core of this mapping are *virtual index MAC addresses* ($M_{v1}, M_{v2}, \ldots$) that are used to reference a peer-specific slot in the L25ARP cache.



Kernel ARP Cache		L25ARP Cache	
IP	Index MAC	Index MAC	Physical MACs
IP$_1$	M_{v1}	M_{v1}	M_{p8} M_{p9}
IP$_2$	M_{v2}	M_{v2}	...

Fig. 5: Sequence diagram of L25ARP for our L25MCP prototype. For an ARP request received from the VETH tunnel, we send out N L25ARP requests and receive up to N L25ARP replies from the network (M_{p8}, M_{p9}). To store this mapping, we create a virtual index MAC (M_{v1}) that we use as an index to the L25ARP cache and that we return to the kernel's ARP implementation. The table contains the ARP tables after this L25ARP exchange.

Since we also communicate this index MAC in ARP replies to the Linux kernel, subsequent packets from the VETH can be directly mapped to a peer slot. Fig. 5 shows the exchange of ARP and L25ARP messages between two nodes with two physical interfaces each (M_{p1}, M_{p2} and M_{p8}, M_{p9}) .

Implementation Details: XDP was originally designed for Ethernet, but a generic mode enables it for any network device in Linux. When the XPM pushes a frame onto a WiFi device via an XDP socket, the kernel's 802.11 layer does not set the WiFi QoS user priority. We solve this with an Extended Berkeley Packet Filter (eBPF) program that modifies the result of the `cfg80211_classify8021d()` kernel function.

By building on XPM, L25MCP is available as a user-space program, which simplifies integration of adaptive, user-defined MC schemes compared to a kernel implementation. Users can also easily include scenario-specific metrics, such as location data, in the decision-making of an MC scheme.

IV. Evaluation

We evaluate our L25MCP prototype in a real-world testbed consisting of two APs and ten stations, each equipped with two WiFi 7 modules. Using WiFi 7 hardware allows us to compare L25MCP directly with MLO on the same radios. Our main questions are whether L25MCP's fine-grained, flexible application of different MC schemes provides measurable

benefits, and whether L25MCP, although being a software implementation, can outperform hardware-based MLO.

A. Evaluation Setup

Our testbed consists of ten client nodes (Dell QCS1250; Intel Core i3 14100) with two Intel BE200 WiFi 7 modules each and a wired server node. We set up all nodes using NixOS 25.11. This provides us with a reproducible software setup. The network is formed by two UniFi U7 Pro APs and a switch. MLO can be enabled or disabled on these APs and we can configure the WiFi bands to be used by MLO. This allows us to configure topologies for MLO and L25MCP as shown in Fig. 1.

As a benchmark, we generate periodic traffic with configurable cycle times, packet lengths, and DSCP value. The traffic is directed from the WiFi nodes to the server and the kernel uses the DSCP value to set the WiFi traffic class. The nodes have Precision Time Protocol (PTP)-synchronized clocks, which the test application uses to measure one-way latency.

We conducted the experiments in the basement of our university's computer science building, where only a few WiFi broadcasts were detectable. To minimize external interference, we selected unused channels in the 5 and 6 GHz bands. We ran each experiment configuration for 120 s and repeated it 100 times.

B. Case Study 1: Firm Real-Time Scenario

First, we want to look at a *firm* [16, p. 2] real-time communication problem: The clients send packets to the server that must arrive within 3 ms. If a packet misses this *deadline* its utility is zero. Reordering is tolerated. This mimics an application that can tolerate up to n consecutive deadline misses before degrading in service quality.

For this experiment, we configure our testbed with three traffic streams: (1) the critical stream sends 200 B every 5 ms, and (2) two background data streams (1,000 B packet length; 10 ms cycle). To get the best results with standard WiFi, we send the critical traffic using WiFi's *voice* access category and the background traffic using the *background* access category. All ten nodes generate these three traffic streams, but we use only one AP because having multiple APs would unfairly benefit L25MCP.

We conduct this experiment for SC (5 GHz and 6 GHz), MLO, and L25MCP (see Fig. 6). For L25MCP, we connect both links to the same channels that are used by MLO in the 5 GHz and 6 GHz bands. While we rely on the MLO implementations of UniFi and Intel, we implement an application-tailored MC scheme with L25MCP for this scenario: For the critical traffic, we perform per-packet Round-robin (RR) LB: Consecutive packets are sent over two different links/channels, reducing the likelihood of losing consecutive packets and help prevent head-of-line blocking by background streams. Each background stream is pinned to a single link.

In Fig. 6, we analyze the number of consecutive deadline misses of the critical traffic stream. The plot shows the cumulative distribution for n consecutive packets that had to be sent until one packet met the 3 ms deadline. The y-axis uses a logarithmic scaling of $1-p$, where p denotes the probability of failure. In the high-reliability regime, the improvement of the residual failure probability is of most importance.

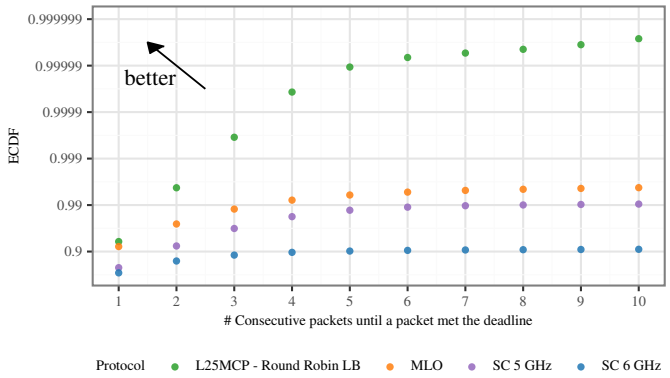


Fig. 6: Probability of reaching the 3 ms deadline with one of n consecutive packets in case study 1.

The SC communication in the 6 GHz band performs the worst. In general, WiFi in the 5 GHz and 6 GHz bands has similar characteristics (unlike the 2.4 GHz band); however, they are not perfectly symmetric, so SC performs differently in these two bands.

With MLO, the residual probability of failure is slightly better than with 5 GHz SC. For example, in 99.22 % of all cases, at least the fourth packet arrives within the deadline, while 5 GHz-SC achieves 98.22 %. The residual error is reduced by a factor of 2.27.

We analyzed the MLO scheduling of the Intel BE200 module using WiFi monitor mode and tcpdump on the AP. To our surprise, we could not observe any LB by MLO in the first ∼9 s of an experiment; all streams were transmitted via the 5 GHz channel. Only after this warm-up period, we could see that packets were balanced between the 5 GHz and 6 GHz channels. The balancing happened after a random number of packets, which we believe to stem from greedily using the earliest TX opportunity (TXOP) on one of the two links. As the initial 9 s delay might be an implementation weakness, we excluded the first 10 s of data from the analysis.

Even though MLO distributes the overall WiFi load on two channels, the residual probability of consecutive packet loss improves considerably less when we compare it with the results of L25MCP. For four consecutive packets, the residual loss with L25MCP shrinks by more than two orders of magnitude in comparison to MLO (factor 211.55). This shows that the tailored *per-packet RR* with L25MCP is better suited for this specific scenario than MLO with *earliest TXOP*. Hence, we conclude that L25MCP's flexibility for applying application-specific MC schemes proves useful. Also, the experiment demonstrates that our software-based L25MCP prototype can provide competitive performance.

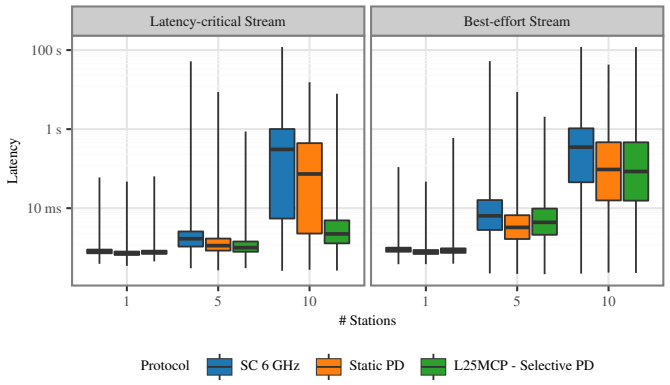


Fig. 7: One-way latencies of different network stream types in case study 2.

C. Case Study 2: Selective Packet Duplication

For the second experiment, we want to demonstrate that L25MCP can utilize multiple and spatially-diversified network links and provide PRP-like redundancy. For this, we mimic an application whose productivity scales directly with the reliability of communication and implement different PD schemes with L25MCP. All clients connect both of their radios to two different 6 GHz channels of two different APs as shown in Fig. 2.

Again, we have a latency-critical traffic stream (5 ms cycle; 200 B packet length) that competes with two best-effort traffic streams (1 ms cycle; 1,000 B packet length) for airtime. As in the first experiment, we utilize WiFi QoS.

For the critical stream, we compare two PD schemes that improve reliability: (1) With *static PD*, we duplicate all traffic and send it to both links, which mimics the behavior of the PRP. (2) With *selective PD*, we only duplicate the critical network stream, the best-effort streams are load balanced on a per-stream basis to individual links. Also, we compare to a SC baseline that sends all streams via one link. In this experiment, we scale the number of clients from one to ten.

Fig. 7 presents the one-way latencies of the second case study, separated by stream type. For a single Client, both PD approaches perform similar to SC, as the wireless channels are far from saturation.

With five clients, both PD schemes provide a small advantage over SC. The 75th percentile of the SC baseline reduces from 2.58 ms to 1.7 ms when static PD is used and to 1.44 ms with selective PD.

With ten clients, the selective PD approach clearly outperforms static PD. It reaches the 75th percentile at 4.91 ms while static PD and SC achieve 442.14 ms and 1.01 s respectively.

Selective PD claims only half of the channel load for background streams when compared to static PD. This has a beneficial impact on the critical stream when there is high channel load.

The latencies of the background streams are slightly better for both PD variants in comparison with SC. With selective PD, this is caused by the reduced overall load on both channels. Static PD gains this advantage by duplication (first of two packets wins) at the cost of negatively impacting the latency-critical stream.

This experiment demonstrates that L25MCP can combine different MC scheme types. We selectively apply PD for critical traffic, which is crucial in order to avoid overloaded channels.

We demonstrated the flexibility and adaptability of L25MCP with two case studies. Still, our application-specific MC schemes do not yet fully exploit the protocol's flexibility as we do not dynamically adapt to unknown application streams or network conditions. Hence, we envision integrating Machine Learning (ML) with L25MCP to dynamically adapt the applied MC scheme on a per-packet basis.

V. Related Work

Enhancing wireless communication using multiple radios has already been investigated when Universal Mobile Telecommunications System (UMTS) was the predominant technology for cellular communication. Dimou et al. [17] propose scheduling strategies for LB and PD using channel metrics as feedback. They discuss a *Generic link layer* that connects one IP stack to multiple L2 devices. Targeting different Radio Access Technologys (RATs), namely UMTS and WiFi, they operate in separate L2 networks. L25MCP allows one host to operate with two links in one WiFi network.

Roman et al. [9] contribute an adaptive algorithm that combines different MC scheduling strategies for use with multiple WiFi links. For integrating their approach into the network stack, they assume a *layer 2.5* on top of multiple WiFi devices without specifying it in detail. L25MCP fills exactly this gap. The same is true for further works on adaptive MC scheduling schemes [12], [13]. For example CIAS [13] provides an algorithm which the authors implement in the application. It uses WiFi's Modulation and Coding Scheme (MCS) as an ISO/OSI layer 1 (L1) input metric. With L25MCP, CIAS can be implemented transparently for the application, in user-space.

The redundancy protocols PRP [3] and FRER [4] have been adopted for the use with WiFi [5], [6]. As discussed in Sec. I, these protocols do not provide other MC schemes than PD and require a redundant network.

Cena et al. [18], [19] improve PRP over WiFi by preventing unnecessary duplicates. They cancel the transmission of a packet if the duplicate has been acknowledged already. When the duplicate is successfully discarded, this approach is effectively a *first TXOP* LB scheduling, as used by MLO. As with MLO, their approach requires implementation in the WiFi module. Also, it is not compatible with WiFi's block acknowledgments. L25MCP is implemented in software and hence independent from closed-source WiFi modules. Also, it works with any recent version of WiFi.

The 802.11 standard introduces MLO in WiFi 7 [7]. MLO provides LB scheduling on the hardware level. Fang et al. [6] propose the integration of FRER with MLO for wireless TSN, which requires a TSN-capable Ethernet network.

For WiFi 8, multiple features that rely on multi-AP coordination have been proposed [20], [21]. One approach provides a different integration of MLO with FRER [22] to realize spatial diversity and PD without relying on a redundant infrastructure. Duplicated frames are forwarded by all APs to a primary AP, who sends the de-duplicated traffic to the target host. Similar to the QoS access categories, they propose per frame redundancy categories.

A WiFi-integrated solution avoids changes on Ethernet hosts, but it relies on being implemented by hardware vendors. For low-cost equipment, this is questionable as such a feature would not be required by the majority of customers. By contrast, the implementation of L25MCP in software makes the protocol hardware agnostic and allows the integration of custom MC scheduling schemes. L25MCP realizes spatial diversity for any (and even mixed) versions of WiFi, at the cost of duplicated traffic on the Ethernet segment of the L2 domain. However, we consider wired bandwidth to be cheap.

VI. Conclusion

We presented L25MCP, an application-transparent Layer 2.5 Multi-connectivity Protocol designed to enhance wireless reliability within a single Layer 2 broadcast domain. By decoupling multi-connectivity scheduling from underlying hardware constraints, L25MCP provides a flexible framework that supports fine-grained, per-packet execution of load balancing, packet duplication, and packet splitting. To resolve multi-interface address mappings, we introduced L25ARP, which seamlessly translates a single IP address to multiple MACs.

Our real-world evaluation in a WiFi 7 testbed confirms that L25MCP translates this structural flexibility into actual performance gains. For firm real-time applications, an application-tailored load-balancing scheme reduces the residual failure probability of consecutive deadline misses by a factor of 212 compared to WiFi 7 MLO. Furthermore, we demonstrate that selective packet duplication can effectively mitigate channel congestion in multi-client scenarios. With our user-space L25MCP prototype, we demonstrate that software-defined multi-connectivity can surpass rigid, hardware-bound solutions and it provides a foundation for more MC experiments.

References

[1] International Federation of Robotics, "A mobile revolution – how mobility is reshaping robotics," International Federation of Robotics, Information Paper, 2021. Accessed: Jun. 20, 2026. [Online]. Available: https://ifr.org/papers/a-mobile-revolution-how-mobility-is-reshaping-robotics

[2] Informationstechnische Gesellschaft im VDE (ITG), "Funktechnologien für industrie 4.0, Positionspapier," Verband der Elektrotechnik Elektronik Informationstechnik e.V. (VDE), Positionspapier, 2017. Accessed: Jun. 20, 2026. [Online]. Available: https://www.vde.com/resource/blob/1635512/acf5521beb328d25fffda9fc6a723501/positionspapier-funktechnologien-data.pdf

[3] "Industrial communication networks – parallel redundancy protocol (PRP)," IEC, Tech. Rep. IEC 62439-3:2021/COR1:2023, 2023, Corrigendum 1 to IEC 62439-3:2021. [Online]. Available: https://webstore.iec.ch/publication/75278

[4] "IEEE standard for local and metropolitan area networks–frame replication and elimination for reliability (FRER)," *IEEE Std 802.1CB-2017*, 2017. DOI: 10.1109/IEEESTD.2017.8091139

[5] M. Rentschler and P. Laukemann, "Performance analysis of parallel redundant wlan," in *ETFA 2012*, 2012. DOI: 10.1109/ETFA.2012.6489647

[6] J. Fang, S. Sudhakaran, D. Cavalcanti, C. Cordeiro, and C. Chen, "Wireless tsn with multi-radio wi-fi," in *CSCN*, 2021. DOI: 10.1109/CSCN53733.2021.9686180

[7] IEEE Standards Association, "IEEE standard for information technology ... Amendment 2: Enhancements for extremely high throughput (EHT)," *IEEE Std 802.11be-2024*, 2025. DOI: 10.1109/IEEESTD.2024.11090080

[8] G. Cena, M. Rosani, and S. Scanzio, "Packet steering mechanisms for mlo in wi-fi 7," in *ETFA*, 2024. DOI: 10.1109/ETFA61755.2024.10710726

[9] C. Roman, P. Ball, and S. Ou, "A shim layer for heterogeneous wireless communications scheduling in intelligent transport systems," in *ISCC*, 2015. DOI: 10.1109/ISCC.2015.7405512

[10] N. Schwarzenberg, A. Wolf, N. Franchi, and G. Fettweis, "Quantifying the gain of multi-connectivity in wireless lan," in *EuCNC*, 2018. DOI: 10.1109/EuCNC.2018.8442457

[11] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-reliable low latency communication using interface diversity," *IEEE Trans. on Communications*, vol. 66, no. 3, 2018. DOI: 10.1109/TCOMM.2017.2771478

[12] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Adaptive multi-connectivity scheduling for reliable low-latency communication in 802.11be," in *WCNC*, 2022. DOI: 10.1109/WCNC51071.2022.9771939

[13] P. Jose, M.-T. Suer, and L. Wolf, "Cias: An application-defined multi-connectivity scheduler for time-sensitive wi-fi systems," in *DCOSS-IoT*, 2025. DOI: 10.1109/DCOSS-IoT65416.2025.00043

[14] 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, 2020. DOI: 10.1109/COMST.2019.2949750

[15] D. Niederprüm, C. Dietrich, and L. C. Wolf, "XDP packet mill: An L2 frame interception framework," in *PEMWN 2025*, 2025.

[16] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*. 1997, ISBN: 0-306-47055-1.

[17] K. Dimou et al., "Generic link layer: A solution for multi-radio transmission diversity in communication networks beyond 3g," in *VTC-2005-Fall. 2005 IEEE*

62nd Vehicular Technology Conf., 2005., vol. 3, 2005. DOI: 10.1109/VETECF.2005.1558226

[18] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "An enhanced mac to increase reliability in redundant wi-fi networks," in *WFCS 2014*, 2014. DOI: 10.1109/WFCS.2014.6837591

[19] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "Dynamic duplicate deferral techniques for redundant wi-fi networks," in *ETFA*, 2014. DOI: 10.1109/ETFA.2014.7005156

[20] I. Val et al., "Wi-fi 8 unveiled: Key features, multi-ap coordination, and the role of c-tdma," *IEEE Communications Magazine*, vol. 63, no. 11, 2025. DOI: 10.1109/MCOM.001.2500091

[21] D. Nunez, P. Imputato, S. Avallone, M. Smith, and B. Bellalta, "Enabling reliable latency in wi-fi 8 through multi-ap joint scheduling," *IEEE Open Journal of the Communications Society*, vol. 6, 2025. DOI: 10.1109/OJCOMS.2025.3549586

[22] G. Cena, S. Scanzio, D. Cavalcanti, and V. Frascolla, "Seamless redundancy for high reliability wi-fi," in *WFCS*, 2023. DOI: 10 . 1109 / WFCS57264 . 2023 . 10144228