

Encapsulation of Industrial Ethernet Protocols over 5G Networks Using L2TPv3 Tunneling

José Luis Maziero Baretta, Nicolas Santos, Christian Mailer,
Adão Boava, Ciro André Pitz and Richard Demo Souza

Abstract—Industry 4.0 demands high reliability and ultra-low latency communications. Although 5G’s Ultra-Reliable Low-Latency Communications (URLLC) meets these requirements, native support for Industrial Ethernet protocols over 5G via Ethernet-type PDU sessions remains unavailable. In this context, this paper proposes and validates an encapsulation technique using the Layer 2 Tunneling Protocol version 3 (L2TPv3) to carry Industrial Ethernet frames over a 5G network emulated with OpenAirInterface and Linux network namespaces. Integration tests confirm successful PROFINET transmission through the 5G RAN, with L2TPv3 encapsulation overhead consistently below 0.25 ms, satisfying the 1 ms URLLC latency budget.

Keywords—5G, Industrial Ethernet, PROFINET, L2TPv3.

I. INTRODUCTION

Industry 4.0 envisions highly automated factories capable of high throughput, determinism, and sub-millisecond latency [1]. The fifth-generation mobile network (5G) addresses these demands through its Ultra-Reliable Low-Latency Communications (URLLC) service category, which guarantees latencies below 1 ms and reliability of 99.999% [2], [3].

Industrial real-time protocols (as PROFINET) rely on Ethernet at OSI Layer 2. Although 3GPP defines Ethernet-type PDU sessions for native Layer-2 transport over 5G [3], this feature remains unavailable for general use due to unresolved MAC-address-based routing and QoS challenges in the User Plane Function (UPF) [4]. This work bridges this gap by encapsulating Industrial Ethernet frames inside L2TPv3 [5] tunnels. The contributions are threefold: (i) a complete emulation environment using OpenAirInterface and Linux network namespaces; (ii) end-to-end validation of PROFINET delivery through the 5G RAN; and (iii) an encapsulation overhead measurement confirming URLLC compliance.

II. BACKGROUND

A. 5G Architecture

The 5G System (5GS) comprises the Next-Generation Radio Access Network (NG-RAN) and the 5G Core (5GC). The NG-RAN consists of gNBs linked to the 5GC via N2 (control) and N3 (user-plane) interfaces [2]. The 5GC adopts an Service-Based Architecture in which virtualized Network Functions (AMF, SMF, and UPF) expose REST APIs over a common service bus [2]. Table I presents the key 5G NR technical requirements defined by 3GPP compared with 4G LTE-A.

J. L. M. Baretta, N. Santos, C. Mailer, A. Boava, C. A. Pitz and R. D. Souza are with the Federal University of Santa Catarina (UFSC), Brazil. E-mails: jose.baretta@grad.ufsc.br, nicolas.fs@grad.ufsc.br, c.mailer@posgrad.ufsc.br, adao.boava@ufsc.br, ciro.pitz@ufsc.br, richard.demo@ufsc.br. This work was partially supported by CNPq INCT STREAM (409179/2024-8).

TABLE I
5G NR TECHNICAL REQUIREMENTS (3GPP / ITU-R IMT-2020).

Parameter	4G LTE-A	5G	Use Case
Peak data rate	500/1000 Mbps	10/20 Gbps	eMBB
Energy efficiency	n/a	Maximum	eMBB
Max UE speed	350 km/h	500 km/h	eMBB
Latency	5 ms	0.5 ms	URLLC
Reliability	n/a	99.999%	URLLC
Battery life	10 years	10–15 years	mMTC
Connection density	60 k/km ²	1 M/km ²	mMTC

B. Industrial Ethernet Protocols

Industrial Ethernet extends IEEE 802.3 for real-time, deterministic manufacturing communication [6]. Protocols span three classes: Class A (e.g., Modbus TCP) uses the standard TCP/IP stack; Class B (e.g., PROFINET RT) bypasses IP, placing payload directly in Ethernet frames; and Class C (e.g., EtherCAT, PROFINET IRT) uses dedicated Layer-2 hardware for the shortest cycle times [3]. Because 5G IP-type PDU sessions strip Layer-2 headers, Class B and C protocols require an encapsulation mechanism.

C. L2TPv3 Tunneling

L2TPv3 (RFC 3931) carries raw Ethernet frames in UDP/IP packets, enabling Layer-2 traffic to traverse Layer-3 networks [5]. The Linux kernel implements it natively via the `ip l2tp` command, exposing virtual Ethernet interfaces (`l2tpethN`) with encapsulation in the kernel data path.

D. Emulation Environment

OpenAirInterface (OAI) is an open-source 5G stack that emulates the full NG-RAN using an RF simulator (`--rfsim`), creating TUN interfaces (`oaitun_enb1`, `oaitun_ue1`) as the IP substrate for L2TPv3 tunnels [7]. Linux network namespaces provide isolated networking environments for each node, gNB, UE, and industrial endpoints, connected via virtual Ethernet (`veth`) pairs, all within a single physical machine [8].

III. PROPOSED SYSTEM

Following the Layer-2 tunneling approach outlined in Section 6.3 of the 5G-ACIA white paper [3], the proposed solution establishes an L2TPv3 tunnel bridges the industrial Ethernet domain with the 5G UPF, allowing Layer-2 Industrial Ethernet frames to traverse the IP-based 5G user plane transparently.

A. Architecture A – Local Tunnel Validation

The first architecture, illustrated in Fig. 1, bypasses the 5G network entirely. Two Linux network namespaces represent the industrial Controller/Master (CM) and the Sensor/Actuator (SA) nodes. The host machine runs the L2 OT-Infrastructure; two L2TPv3 tunnels connect to the respective namespaces, and a software bridge joins both tunnel endpoints so that frames sent from CM arrive at SA and vice versa.

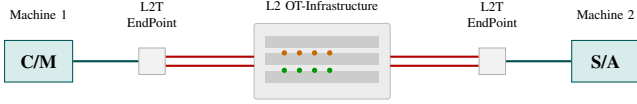


Fig. 1. Proposed architecture for 5G network abstraction and L2TPv3 tunnel verification.

The purpose of this architecture is to validate the encapsulation and decapsulation behavior before introducing the 5G stack. A PROFINET RT packet is crafted with Scapy [9] and injected into the `l2tpeth0` interface of the CM namespace; successful reception at the SA namespace interface confirms tunnel operation.

B. Architecture B – 5G-Integrated Tunnel

Architecture B, illustrated in Fig. 2, extends Architecture A by inserting the 5G radio access link between the two L2TP tunnel endpoints. The gNB is isolated in its own network namespace and launched with OAI (`nr-softmodem --rfsim --phy-test --noS1`). The UE runs on the host and connects to the gNB via the RF simulator, creating the `oaitun_ue1` and `oaitun_enb1` TUN interfaces that carry user-plane IP traffic.



Fig. 2. 5G-Integrated architecture using OAI RF simulator for L2TPv3 tunnel validation.

One L2TPv3 tunnel is established between the host UE endpoint and the gNB namespace over the OAI TUN interface pair; a second tunnel connects the host to the CM namespace over a veth pair. A Linux bridge joins both tunnels, so Industrial Ethernet frames originating in the CM namespace are encapsulated, forwarded through the 5G radio emulator, and delivered to the gNB namespace.

At the time of writing, the OAI 5G emulator supported only a single UE per simulation run. Consequently, the full topology ($CM \leftrightarrow 5G \leftrightarrow SA$) was simplified to $CM \leftrightarrow 5G \leftrightarrow gNB$. This simplification is valid because arrival of the Industrial Ethernet frame at the gNB already confirms correct encapsulation and over-the-air transmission; the reverse path is symmetric by construction at the L2TPv3 and IP layers.

IV. EXPERIMENTAL METHODOLOGY

All experiments were conducted on a single host running Ubuntu 20.04 LTS with Linux kernel 5.4, an Intel Core i7 processor at 3.6 GHz, and 16 GB RAM.

A. Correctness Testing

A PROFINET RT Class B packet (FrameID `0x8000`) was crafted using the Scapy network packet manipulation framework [9]. The packet structure included a `PNIORealTimeCyclicPDU` carrying two data slots with test pattern AAA and a PROFIsafe block. The packet was injected into the L2TPv3 interface inside the CM namespace; `tcpdump` running on the far-end interface confirmed reception with matching payload (`0x8892` Ethernet type and AAA pattern). This test was repeated for both Architectures A and B.

B. Encapsulation Overhead Measurement

Measuring absolute latency in a purely software-emulated environment is unreliable because the host OS scheduler introduces significant jitter. Instead, encapsulation overhead is isolated as the differential between the mean transport time of encapsulated and non-encapsulated packets of the same size traversing an identical software path (veth pair without L2TP).

A Python/Scapy sender injects 500 packets per size step, sweeping payload sizes from 50 bytes to 1400 bytes in 50-byte increments. The MTU of each L2TPv3 interface is set to 1446 bytes to accommodate the L2TP/UDP/IP header overhead while staying within standard Ethernet MTU.

Each packet embedded a Unix timestamp in its payload; the receiver recorded the kernel arrival timestamp. Transport time is computed as

$$\Delta t = t_{\text{arrival}} - t_{\text{send}}. \quad (1)$$

The mean over 500 samples is used for each size point. All measurements are obtained for Architecture A; Architecture B results are validated qualitatively through `tcpdump` packet capture. Moreover, the encapsulation overhead is

$$\Delta e = \Delta t_{\text{encap}} - \Delta t_{\text{no-encap}}. \quad (2)$$

V. RESULTS AND DISCUSSION

A. Functional Validation

In Architecture A, the `tcpdump` capture on the SA namespace interface confirmed reception of the PROFINET frame with identical payload to the injected packet, verifying correct L2TPv3 encapsulation and decapsulation.

Regarding Architecture B, an identical PROFINET frame was injected from the CM namespace and captured at the `l2tpeth0` interface of the gNB namespace. The Ethernet type field (`0x8892`) and payload matched exactly, confirming that the frame traversed the L2TPv3 tunnel, crossed the OAI 5G RF-simulated radio link, and arrived at the gNB.

The identical payload across both captures, including the AAA test pattern, confirms end-to-end Layer 2 transparency through the 5G-integrated tunnel. Table II summarizes the functional validation outcomes for both architectures.

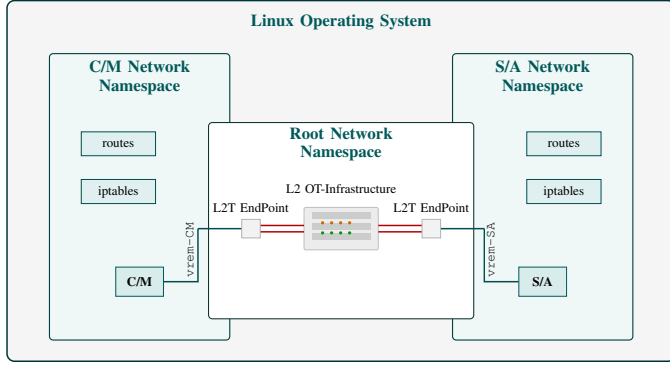


Fig. 3. Detailed Architecture A: Linux namespaces, L2TPv3 tunnel endpoints, and software bridge configuration.

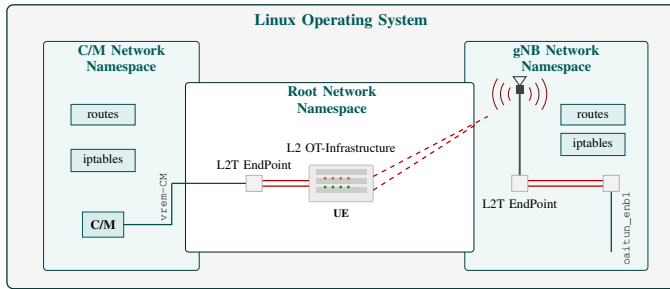


Fig. 4. Detailed Architecture B: CM and gNB namespaces connected via OAI RF simulator TUN interfaces and L2TPv3 tunnel.

B. Encapsulation Overhead

Figure 5 shows the mean transport time as a function of packet size for encapsulated packets, non-encapsulated packets, and the differential (encapsulation overhead). The overhead Δe remained below 0.25 ms across all tested packet sizes, keeping the RTT contribution ($2 \times \Delta e < 0.5$ ms) well within the 1 ms URLLC budget. The overhead was largely independent of packet size, consistent with L2TPv3's fixed-size header (~ 50 bytes), where processing is dominated by header construction rather than data copy. Residual jitter is attributed to OS scheduler interference and would be substantially reduced in a hardware deployment with dedicated NICs and real-time kernel patches.

C. Limitations and Future Work

Two limitations remain. First, L2TPv3's fixed ~ 50 -byte header overhead may be disproportionate for bandwidth-constrained applications with very small payloads, where 5G-native IoT protocols would be more appropriate. Second, the OAI emulator's single-UE restriction prevented full symmetric topology validation (CM $\leftrightarrow$ 5G $\leftrightarrow$ SA). Future work will

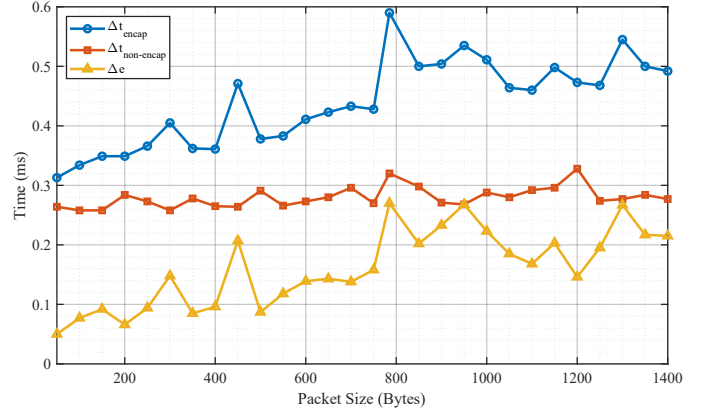


Fig. 5. Mean transport vs. packet size for encapsulated (Δt_{encap}) and non-encapsulated ($\Delta t_{\text{non-encap}}$) packets, and encapsulation overhead Δe .

address this using OAI multi-UE mode or a real 5G testbed, alongside experiments with hardware-based PROFINET controllers and realistic 5G channel models.

VI. CONCLUSION

This paper demonstrated the viability of encapsulating Industrial Ethernet protocols over 5G networks using L2TPv3. Results obtained with OpenAirInterface showed an encapsulation overhead consistently below 0.25 ms, suggesting that this stage of the processing chain is compatible with the URLLC latency budget. By relying on Linux kernel networking, this solution enables the seamless integration of legacy industrial equipment into 5G-enabled factories without proprietary hardware. Future work will focus on physical testbed validation and the assessment of scalability with multiple UEs.

ACKNOWLEDGEMENTS

The authors would like to thank CNPq INCT STREAM (409179/2024-8) for the financial support.

REFERENCES

- [1] 3GPP, "TS 22.104: Service requirements for cyber-physical control applications in vertical domains," Tech. Rep. Version 18.3.0, Release 18, 3GPP, May 2024.
- [2] C. Cox, *An Introduction to 5G: The New Radio, 5G Network and Beyond*. Wiley, 2020.
- [3] 5G ACIA, "Integration of industrial ethernet networks with 5G networks," white paper, 5G ACIA, 2019.
- [4] O. Adamuz-Hinojosa, F. Delgado-Ferro, J. Navarro-Ortiz, P. Muñoz, and P. Ameigeiras, "Unleashing 5G seamless integration with TSN for industry 5.0: Frame forwarding and QoS treatment," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 4874–4884, May 2025.
- [5] J. Lau, M. Townsley, and I. Goyret, "Layer two tunneling protocol – version 3 (L2TPv3)," RFC 3931, Internet Engineering Task Force, Mar. 2005.
- [6] Siemens, *Industrial Communication: SIMATIC NET Industrial Ethernet Networking Manual*. Siemens AG, Nuremberg, Germany, 2008. Order No.: 6GK1970-1BA10-0AA0.
- [7] F. Völk, T. Schlichter, F. Kaltenberger, T. Heyn, G. Casati, R. T. Schwarz, and A. Knopp, "Field trial of a 5G non-terrestrial network using OpenAirInterface," *IEEE Open J. Veh. Technol.*, vol. 3, pp. 243–250, 2022.
- [8] Linux Kernel Organization, "namespaces(7) - linux programmer's manual." <https://man7.org/linux/man-pages/man7/namespaces.7.html>, 2022. Accessed: May 25, 2026.
- [9] P. Biondi, "Scapy: interactive packet manipulation program." [Online]. Available: <https://scapy.net/>, 2024.

TABLE II

FUNCTIONAL VALIDATION SUMMARY FOR ARCHITECTURES A AND B.

Arch.	Injection	Capture	Ethertype	Payload	Result
A	CM ns	SA ns	0x8892	AAA	PASS
B	CM ns	gNB ns	0x8892	AAA	PASS