ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The Future of Industrial Networking: Embracing TSN Protocols in the Wireless Era

Andrea Garbugli, Armir Bujari

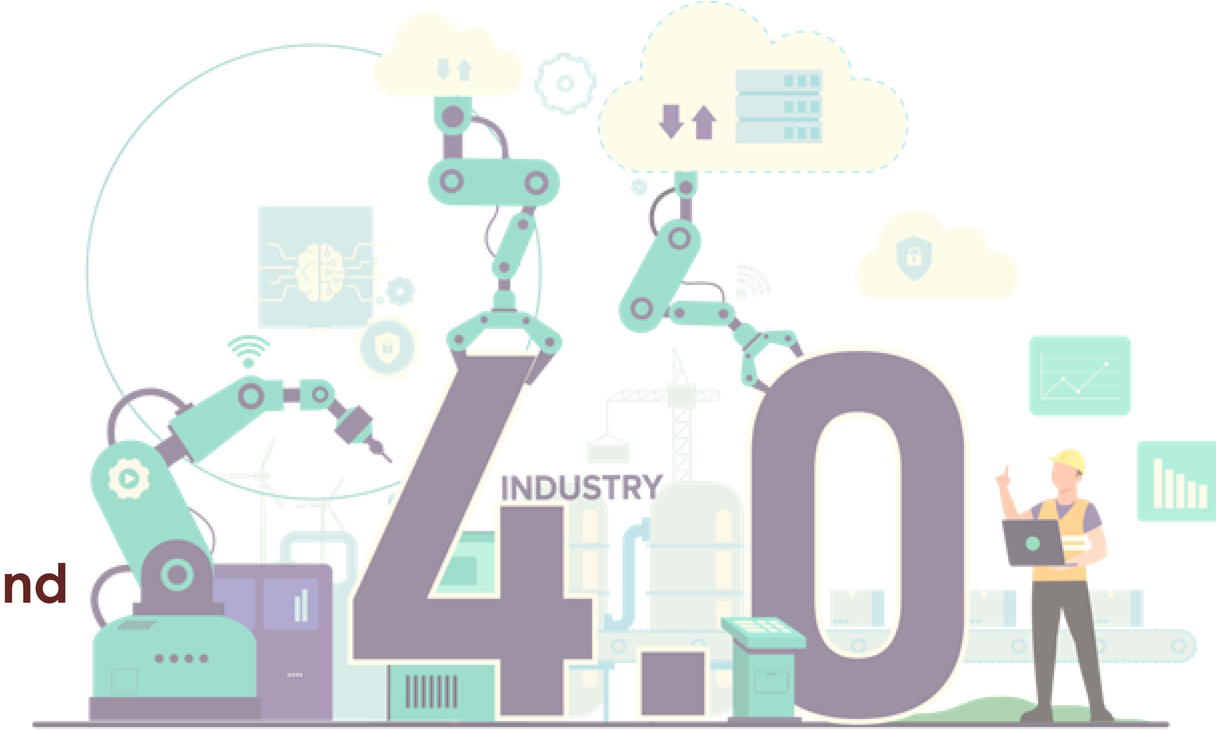
Department of Computer Science and Engineering (DISI)
University of Bologna – Italy

Industry 4.0

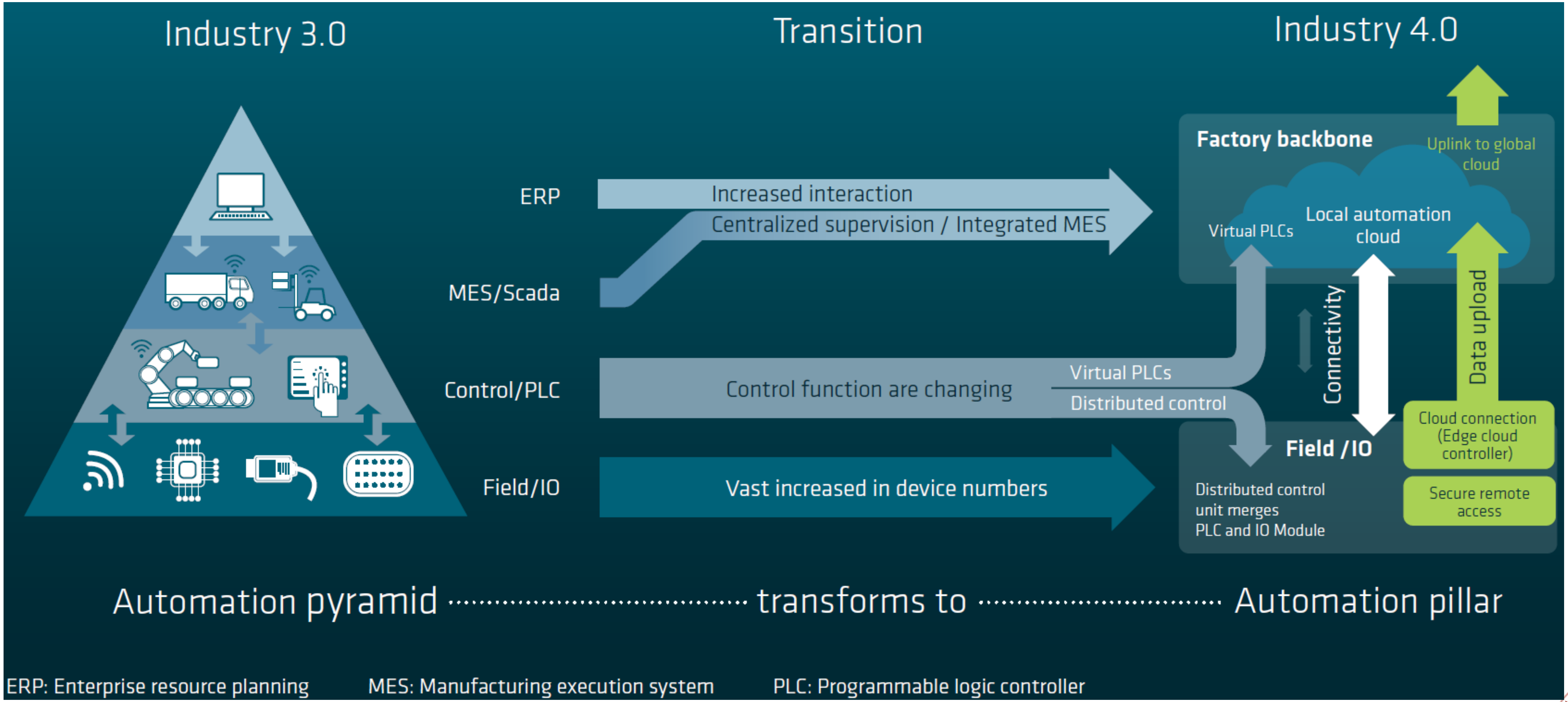
A large spread phenomenon and trend to consider an evolution of traditional **industrial processes**

Industry 4.0 (I4.0) has multiple meanings:

- Connects/merges **production with ICT**
- Merges **customer data with machine data**
- Goes **M2M**: Machines communicate with machines
- Components and machines autonomously manage **production in a flexible, efficient, and resource-saving manner**



From Industry 3.0 to Industry 4.0



Industry 4.0: Technological Enablers

1	Advanced Manufacturing Solutions	• Autonomous, cooperating industrial robots • Numerous integrated sensors and standardized interfaces
2	Additive Manufacturing	• 3D printing, particularly for spare parts and prototypes • Decentralized 3D facilities to reduce transport distances and inventory
3	Augmented Reality	• Augmented reality for maintenance, logistics, and all kinds of SOP • Display of supporting information, e.g., through glasses
4	Simulation	• Simulation of value networks • Optimization based on real-time data from intelligent systems
5	Horizontal/ Vertical Integration	• Cross-company data integration based on data transfer standards • Precondition for a fully automated value chain (from supplier to customer, from management to shop floor)
6	Industrial Internet	• Network of machines and products • Multidirectional communication between networked objects
7	Cloud	• Management of huge data volumes in open systems • Real-time communication for production systems
8	Cyber-security	• Operation in networks and open systems • High level of networking between intelligent machines, products, and systems
9	Big Data and Analytics	• Full evaluation of available data (e.g., from ERP, SCM, MES, CRM, and machine data) • Real-time decision-making support and optimization

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Industrial Communication Requirements

Service requirements range from **best-effort traffic** to **critical real-time traffic**

Several organizations (e. g., 3GPP, IEC, IEEE, IIC) have defined **traffic types** and corresponding **requirements** of relevance to **industrial automation**

Needs for **appropriate QoS mechanisms** for the application's data transmission

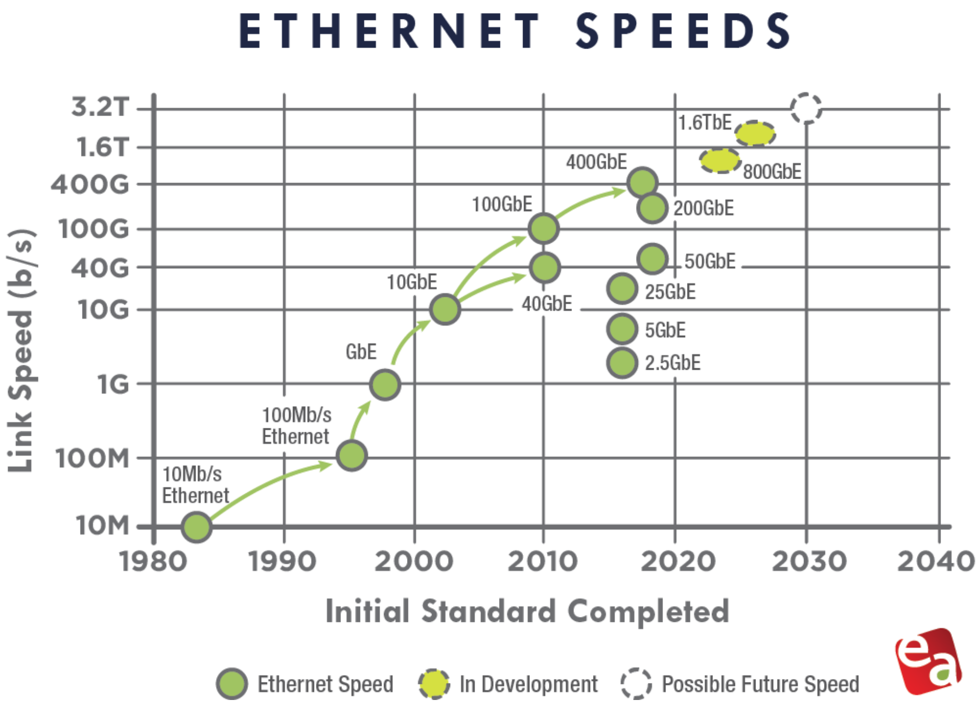
Need for (and **convergence of**) **wireless** & **wired** communication technologies

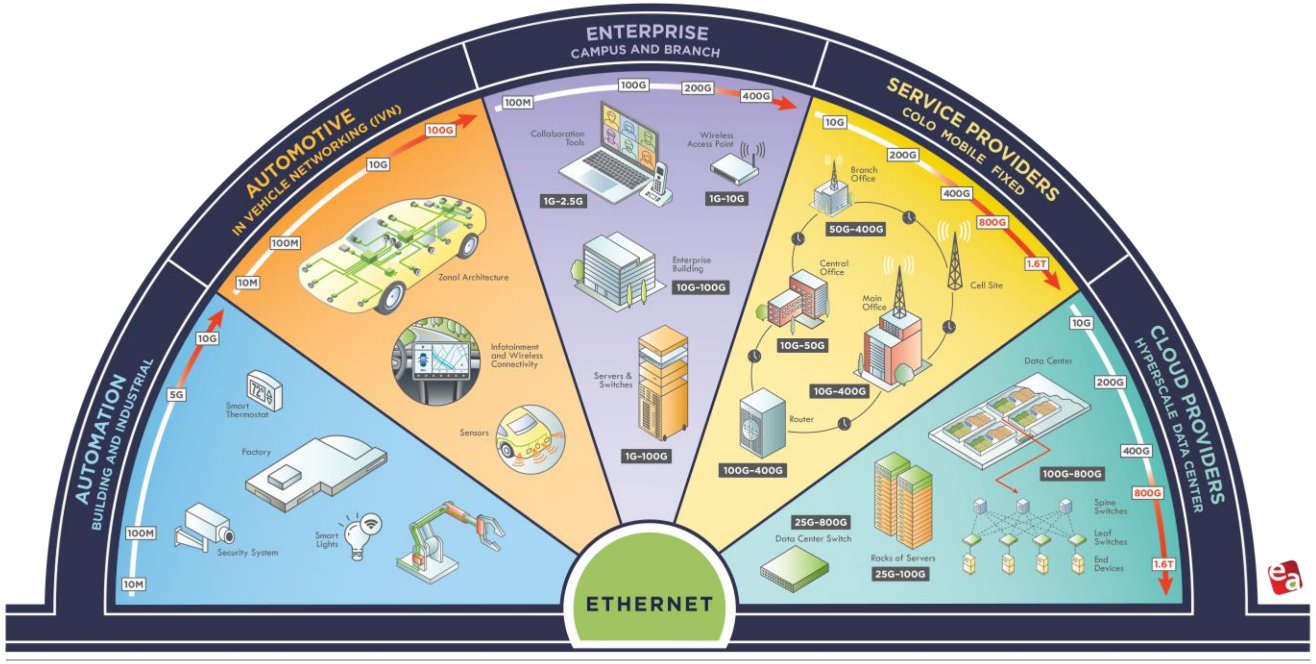
Table 1: Industrial automation traffic types, service requirement and related TSN features [33] [14]

Traffic types	Periodic / Sporadic	Typical period	Data delivery guarantee	Tolerance to Jitter	Tolerance to loss	Typical data size (Byte)	Criticality
Isochronous	P	100 µs ~ 2 ms	Deadline	0	None	Fixed: 30 ~ 100	High
Cyclic -Synchronous	P	500 µs ~ 1 ms	latency bound (τ)	≤ τ	None	Fixed: 50 ~ 1000	High
Cyclic -Asynchronous	P	2 ms ~ 20 ms	latency bound (τ)	≤ τ	1 ~ 4 Frames	Fixed: 50 ~ 1000	High
Events: control	S	10 ms ~ 50 ms	latency bound (τ)	n.a.	Yes	Variable: 100 ~ 200	High
Events: alarm & operator com-mands	S	2 s	latency bound (τ)	n.a.	Yes	Variable: 100 ~1500	Medium
Network control	P	50 ms ~ 1 s	throughput	Yes	Yes	Variable: 50 ~ 500	High
Configuration & diagnostics	S	n.a.	throughput	n.a.	Yes	Variable: 500 ~ 1500	Medium
Video	P	Frame Rate	throughput	n.a.	Yes	Variable: 1000 ~ 1500	Low
Audio/Voice	P	Sample Rate	throughput	n.a.	Yes	Variable: 1000 ~1500	Low
Best effort	S	n.a.	None	n.a.	Yes	Variable: 30 ~ 1500	Low

Ethernet Rules Industrial Networked Environments

Ethernet has become a new **standard for future industrial** and **automation applications**, surpassing Fieldbus technologies





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Networking Enabler (1/3): Time-Sensitive Networking (TSN)

Many ***Industrial* Ethernet** variants, e.g., PROFINET and EtherCAT. Most of them suffer **incompatibility problems**

IEEE introduced a suite of standards called ***Time-Sensitive Networking (TSN) a.k.a Deterministic Ethernet***

IEEE Standards	Title	Description
IEEE 802.1AS, IEEE 1588	Timing Synchronization for Time-Sensitive Applications	Specialized version of the generic Precision Time Protocol (gPTP) to synchronizes clocks between network devices
IEEE 802.1Qbv	Enhancements to Traffic Scheduling Time-Aware Shaper (TAS)	Enables Ethernet frames to be transmitted on a schedule (guaranteed), while allowing [non-] time-sensitive frames to be transmitted on a best-effort basis (no guarantee). Each frame is assigned a queue based on QoS priority
IEEE 802.1Qbu	Frame Preemption	Enables frame pre-emption to interrupt the transmission of frames in favor of high priority frames
IEEE 802.1CB	Frame Replication and Elimination for Reliability	Provides for capabilities to recover from dropped Ethernet frames or broken switches in a TSN network by inserting duplicating frames at the sender and then discarding the duplicate
IEEE 802.1Qat	Stream Reservation Protocol (SRP)	Specifies the admission control framework for admitting or rejecting flows based on flow resource requirements and the available network resources.
IEEE 802.1Qav	Forwarding and Queuing of Time-Sensitive Streams.	Specifies bridge operations that provide guarantees for time-sensitive lossless real-time audio/video (A/V) traffic (i.e. bounded latency and jitter).

Other protocols:

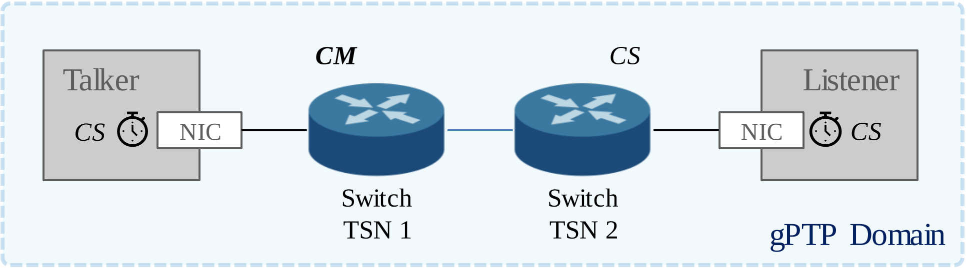
- IEEE 802.1Qcc: Enhancements and Performance Improvements
- IEEE 802.1Qci: Per Stream Filtering and Policing
- IEEE 802.1Qch: Cycling Queuing and Forwarding

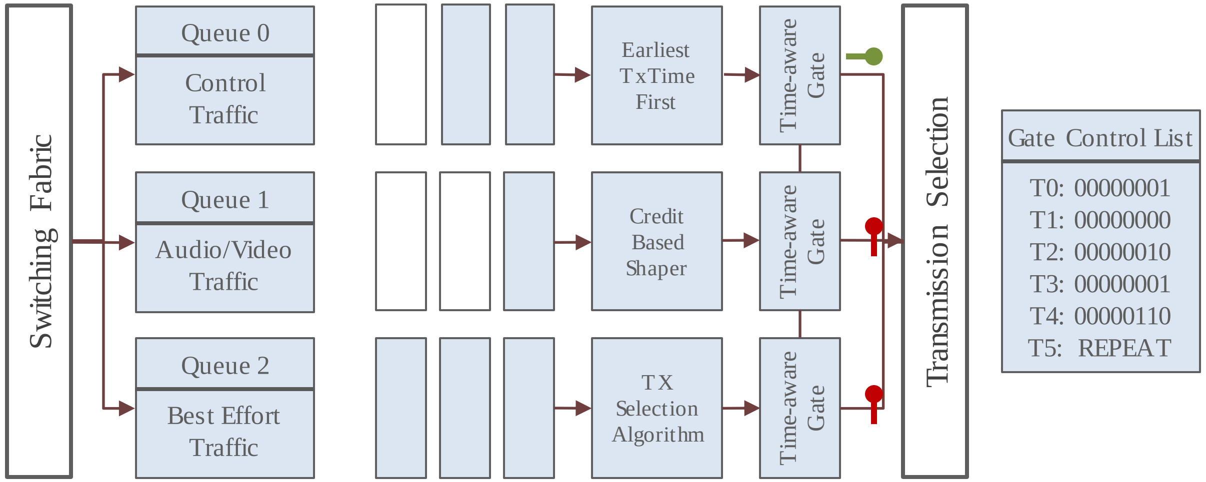
Networking Enabler (1/3): Synchronization and Packet Scheduling in TSN

A set of standards that make **Ethernet networks deterministic**, to support real-time industrial traffic

- Synchronization (IEEE 802.1AS) via a specific profile of the ***Precision Time Protocol*** (PTP): generic PTP (gPTP) [1]
  - ***Clock Master*** (CM), selected during the election phase
  - ***Clock Slave*** (CS)

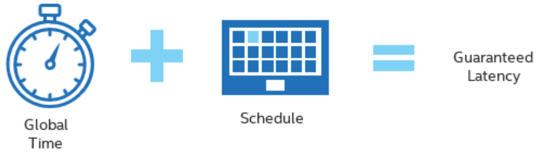
- Enhancements to Traffic Scheduling ***Time-Aware Shaper*** (TAS) [1]
  - Algorithms for selecting the packet to be sent and ***Gate Control List*** (GCL) to create cyclical ***Time-aware Windows***
  - Each frame is assigned a ***queue based on QoS priority***





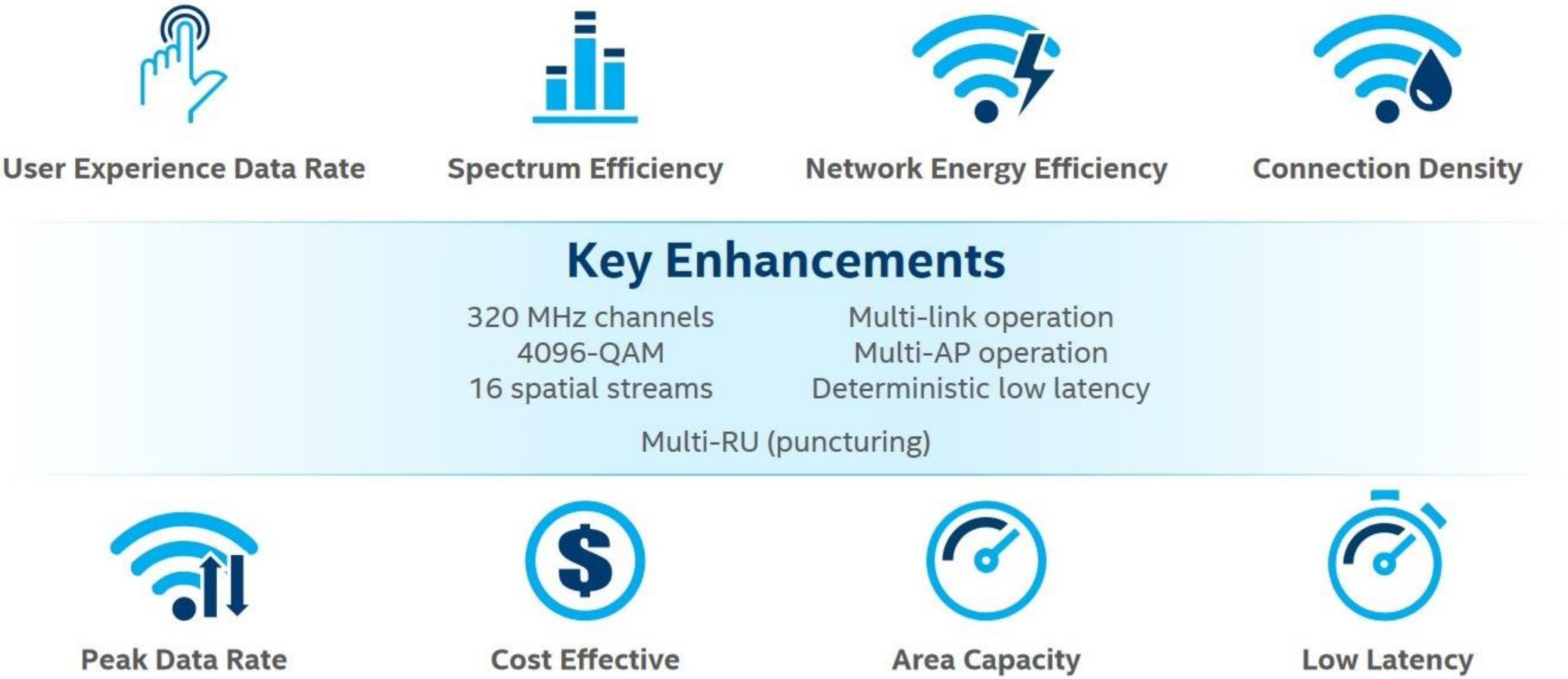
TSN REQUIRES A NIC THAT SUPPORTS

- ***Hardware clock*** → *Precise* ***synchronization***
- ***Multi-queues*** → ***Traffic classes*** associated to NIC queue



[1] Nasrallah, Ahmed, et al. "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research." IEEE Communications Surveys & Tutorials 21.1 (2018): 88-145.

Networking Enabler (2/3): Upcoming Industrial Wi-Fi (IEEE 802.1be – Wi-Fi 7)

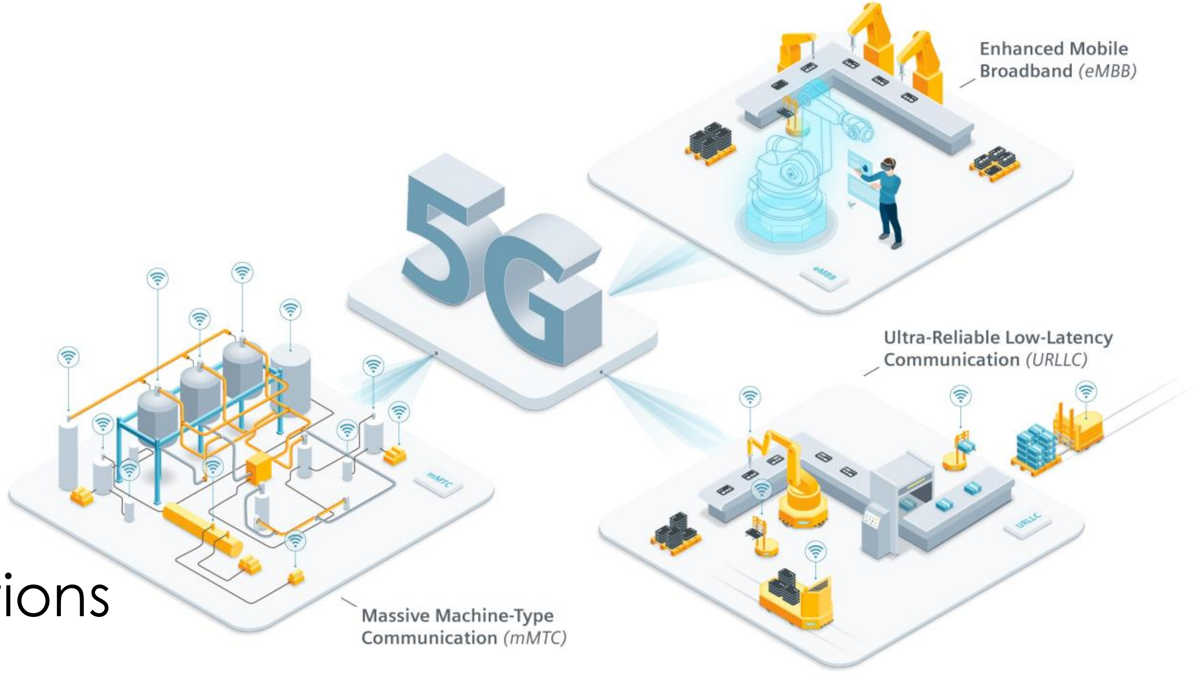


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Networking Enabler (3/3): 5G

5G revolutionizes connectivity beyond 4G with **three key use cases**:

- Enhanced Mobile Broadband **(eMBB)**
- Massive Machine Type Communications (**mMTC**)
- Ultra-Reliable Low-Latency Communications (**URLLC**)



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

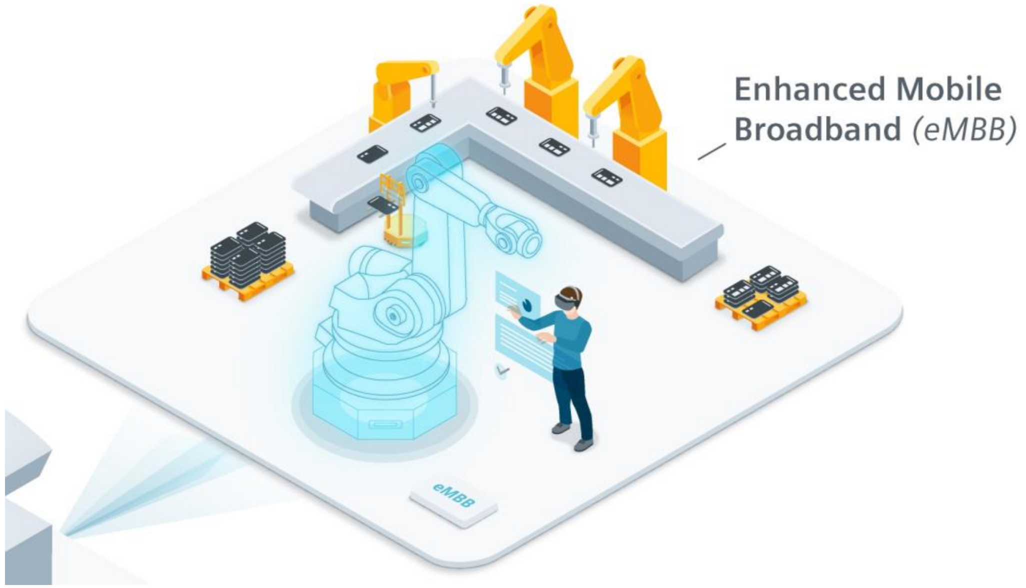
5G Use Cases

Enhanced Mobile Broadband (eMBB)

OBJECTIVE: Efficient communication across a vast number of devices

APPLICATIONS: HD video streaming, virtual/augmented reality, and large data downloads

KEY FEATURES: High data rates, improved capacity, and enhanced connectivity in densely populated areas



- $1000 \times$ Capacity/km^2
- >10 Gbps Peak
- 100 Mbps for Every User
- Spectrum Efficiency

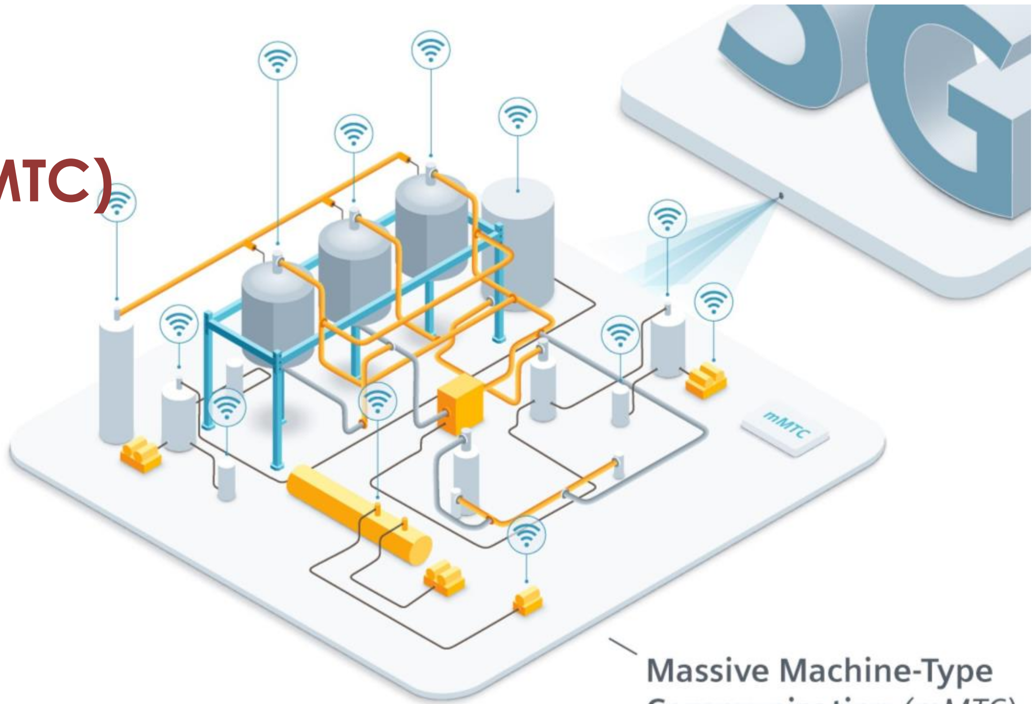
5G Use Cases

Massive Machine Type Communications (mMTC)

OBJECTIVE: High-speed internet access for data-intensive applications

APPLICATIONS: IoT networks, smart cities, environmental monitoring

KEY FEATURES: Low power usage, high scalability, and support for many low-throughput devices



Massive Machine-Type Communication (*mMTC*)

- Sporadic Access
- Energy Optimized (10yr)
- Signaling Reduction
- 1000 × Connected Devices

5G Use Cases

Ultra-Reliable Low-Latency Communications (URLLC)

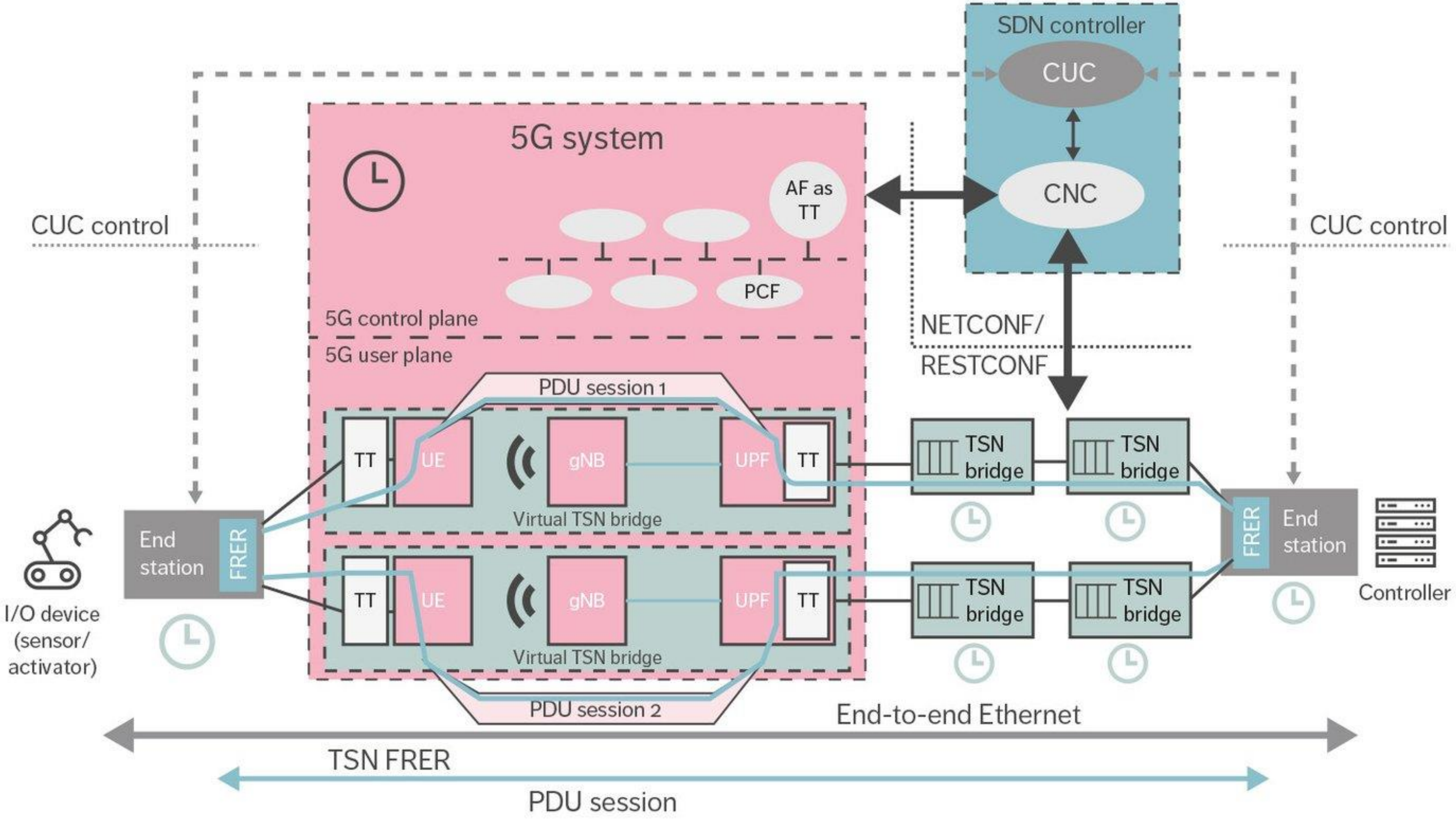
OBJECTIVE: Provide reliable and instant communication for critical applications

APPLICATIONS: Autonomous vehicles, remote surgery, industrial automation, and emergency response

KEY FEATURES: Ultra-low latency, high reliability, and immediate data transfer

- Low Latency (< 1ms)
- High Reliability (99.99999%)
- High Availability
- Reduce Cost per bit

The Future foresees an Integrated Industrial Computing Environment



[1] 5G TSN - integrating for industrial automation - Ericsson

[2] 5G-ACIA, White Paper Integration of 5G with Time-Sensitive Networking for Industrial Communications, 2021

The Computing Continuum and the Next-generation Applications

Vast landscape of **application domains**: Industrial IoT, Vehicular Networks, Smart Cities, AI, and VR/AR.

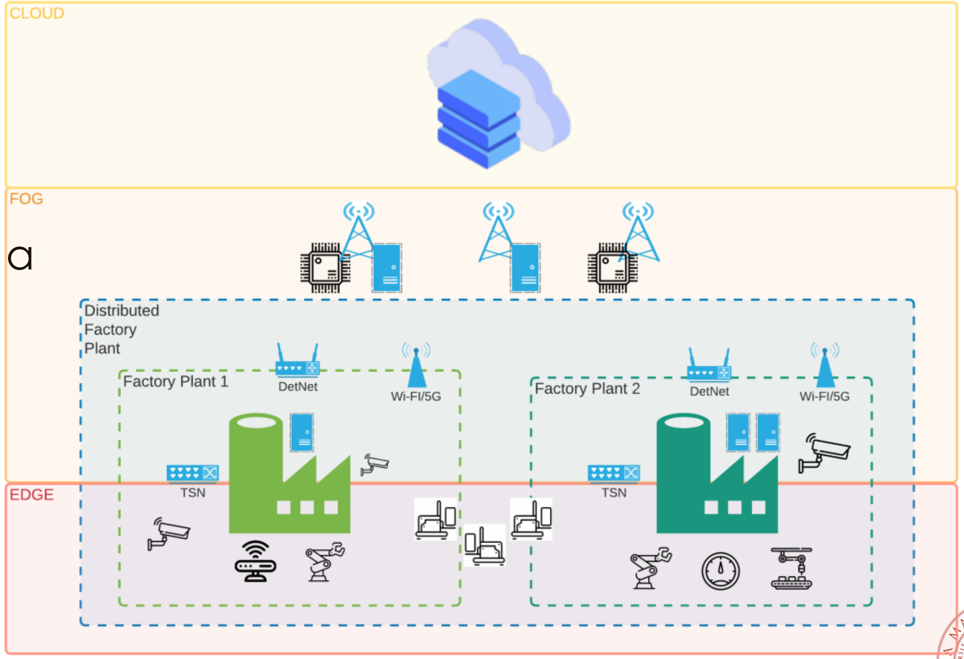
Coexistence of applications with very different **QoS requirements**:

- High throughput,
- Ultra-low latency,
- Deterministic communication,
- and more…

The **Computing Continuum** takes center stage, representing a more **fluid and adaptive cloud environment**

- Heterogeneous **Physical** and **virtual** resources
- Heterogeneous communication technologies

RAN: Radio Access Network
RAN-RT: Real Time Resources
RAN-NRT: Non-Real Time Resources
CP: Control Plane
UP: User Plane



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INSANE:
A Unified Middleware for
QoS-aware Network Acceleration

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Statement (1/2): Evolution Towards Specialized Hardware Solutions

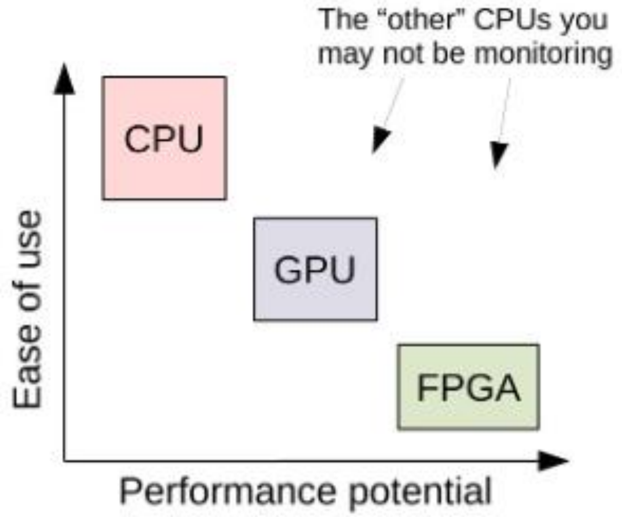
- The network devices landscape has **evolved** to cater the demands of **intensive distributed data applications** and **time-sensitive tasks.**

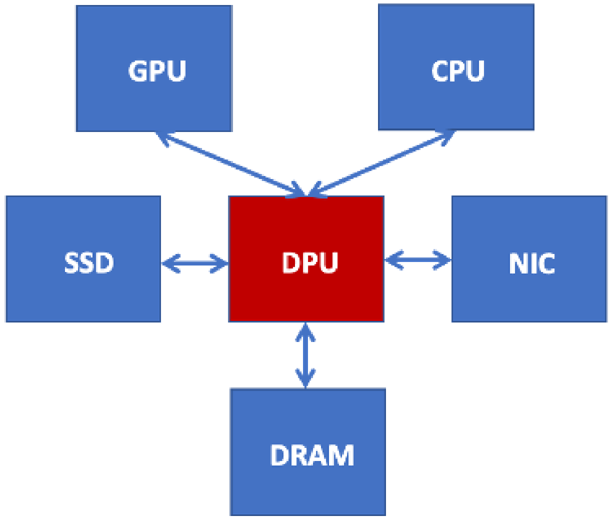
- Unprecedented **network speed**: from Gbps to offering hundreds of Gbps
  → also maintaining **low latency profiles**

- Moving towards a **data-centric network architecture**

- **DPU** for **efficient packet/data processing**.
  - **GPU** to accelerate for **Machine Learning tasks**.
  - **CPU free** for **compute (**application**) execution**.

- **However,** these technologies present **heterogeneous APIs and low-level primitives adding to the complexity**.





Data-centric network architecture

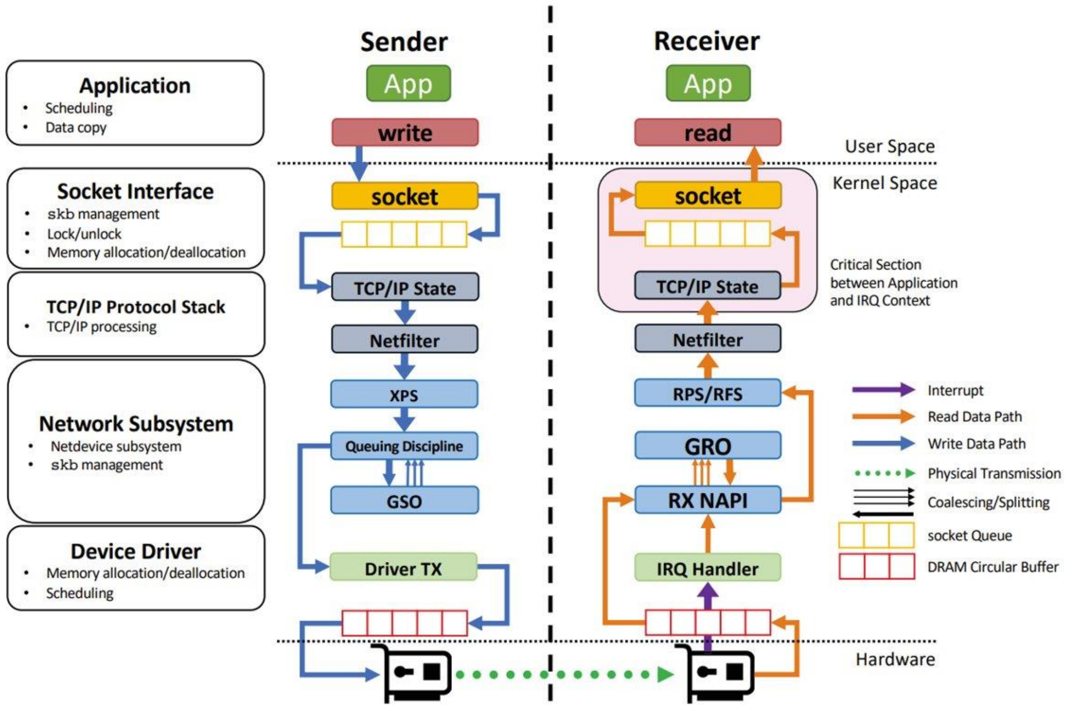
Statement (1/2): Overhead in the (Linux) Software Networking Stack

Linux Networking Stack:
- Robust and efficient* networking infrastructure.
- **Integrated** into the Linux kernel, providing **essential networking services**.
- Enables **communication between devices** over various **network protocols**.

Sockets API:
- **Abstraction layer** for network communication.
- Provides a **unified interface** for applications to interact with the networking stack.
- Facilitates communication over TCP/IP, UDP, and other protocols.

Various sources of **overhead**:
- **Data Copies**
- **Context Switching**

→ impossible to fully utilize the network bandwidth or achieve ultra-low latency supported by modern hardware technologies



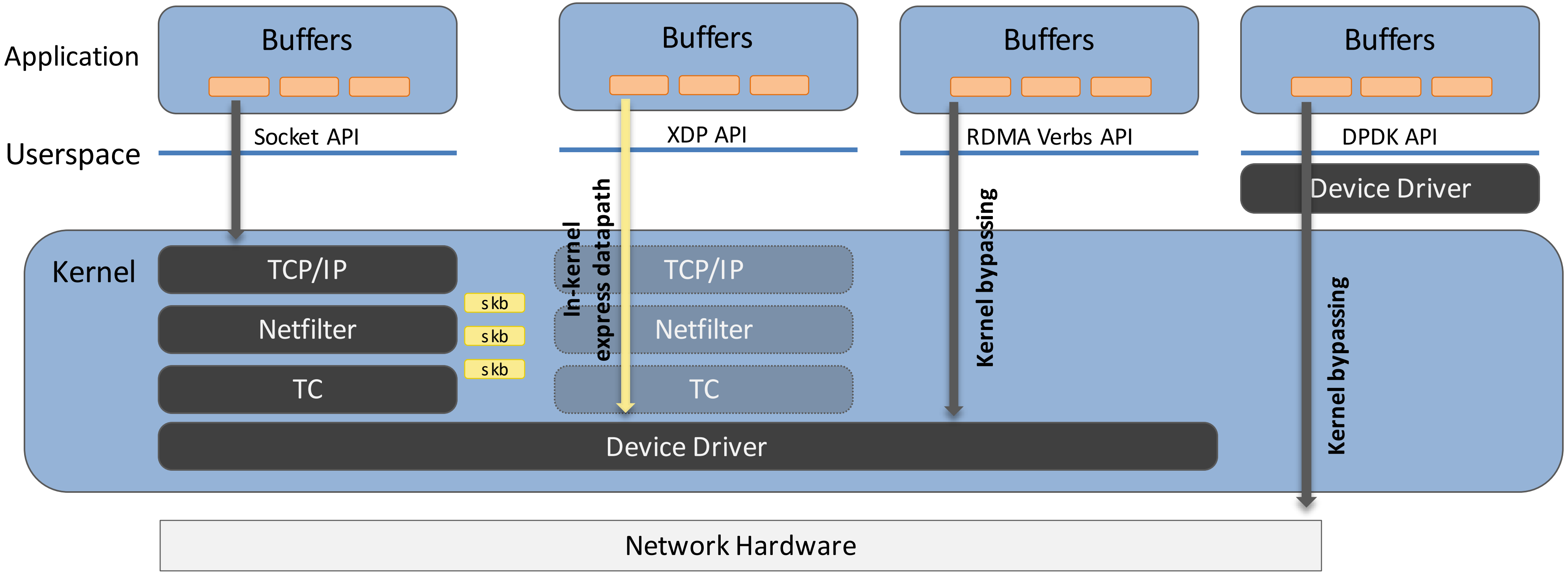
Linux network stack overheads [1]

[1] Cai, Qizhe, et al. "Understanding host network stack overheads." *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 2021.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Hardware and Software Acceleration Technologies

Novel **acceleration options** and **kernel bypassing techniques**: e.g., **RDMA**, **XDP**, **DPDK**.



Different APIs and hardware support between technologies…

INSANE: INtegrated Selective Acceleration at the Network Edge

… Similarities between the different approaches to achieve network acceleration

- **SOLUTION**: An **edge-cloud data distribution middleware** offering developers to selectively accelerate critical parts of their applications

- Associate different **data flows** to different Quality of Service (**QoS**) levels…

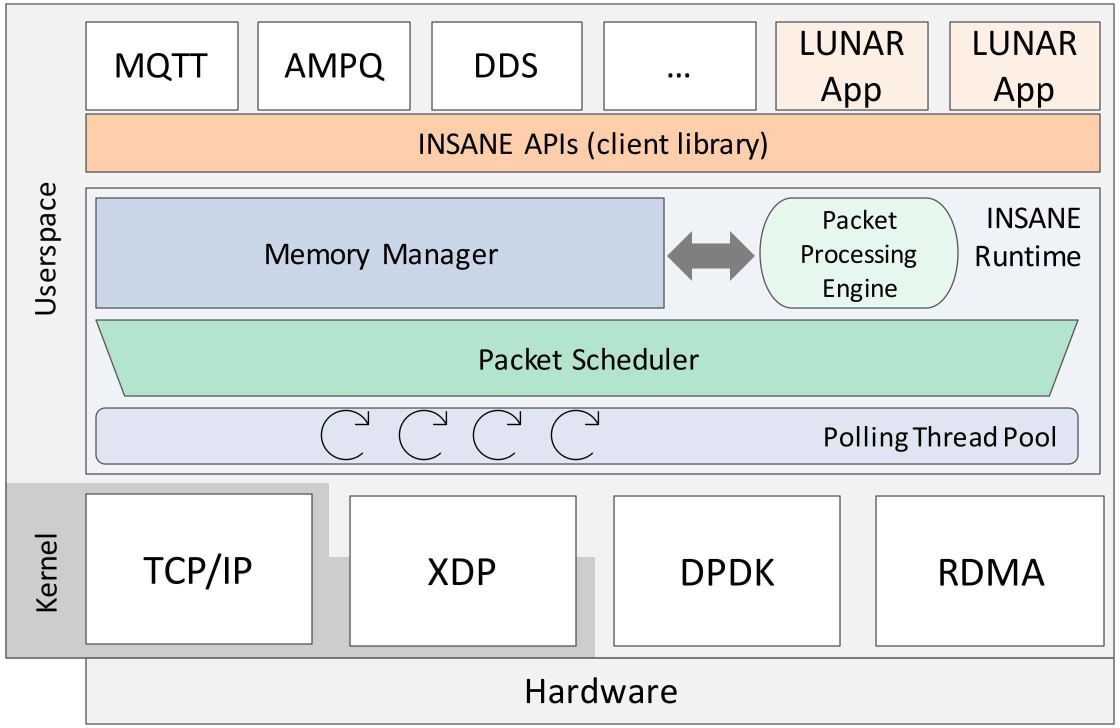
  …direct mapping to the most appropriate network acceleration technology

- Innovative contributions:

  - A **uniform API** for data distribution, based on the **data stream abstraction**

  - **Technology-independent framework** for memory management, zero-copy transfers and efficient packet processing

  - User-space scheduler



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INSANE: Latency and Throughput Evaluation Results and Comparison

Two nodes directly interconnected
→ to minimize **network operations overhead**

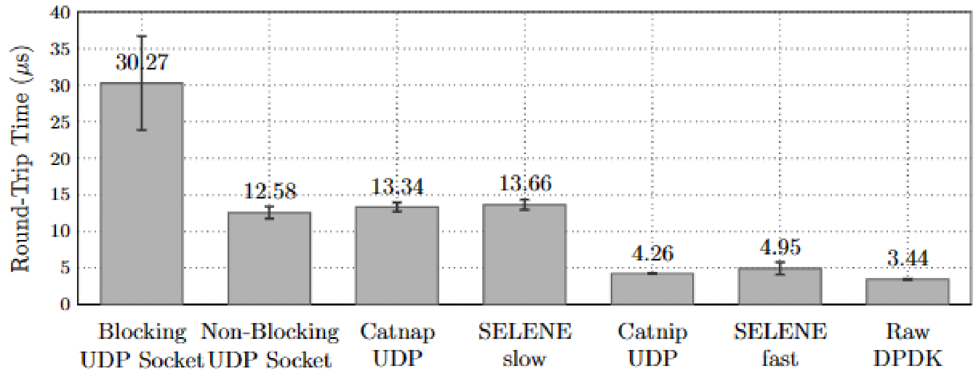
- **OS**: Ubuntu 22.04
- **CPU**: 18-core Intel I9-10980XE @ 3.00Ghz
- **RAM**: 64GB
- **NIC**: Mellanox DX-6 100Gbps

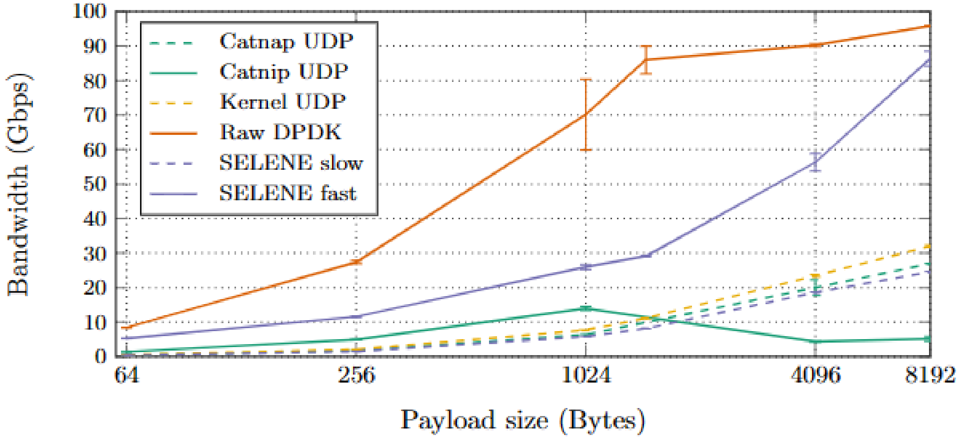
Evaluation running a ping-pong application for the RTT, and one-way source to sink for the Throughput.

Comparison with other State-of-the-Art solution: **Demikernel** [1] (*Catnip* and *Catnap*)

- INSANE has a slightly higher latency (**ns-scale**), but…
- **… Elevated Throughput**

INSANE achieves a good balance between high throughput and low latency.





[1] Zhang, Irene, et al. "The Demikernel Datapath OS Architecture for Microsecond-scale Datacenter Systems." *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*. 2021.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INSANE: Comparison between Insane-based MOM (Lunar), DDS and ZeroMQ

Lunar MOM is implemented on top of INSANE

Two nodes directly interconnected
→ to minimize **network operations overhead**

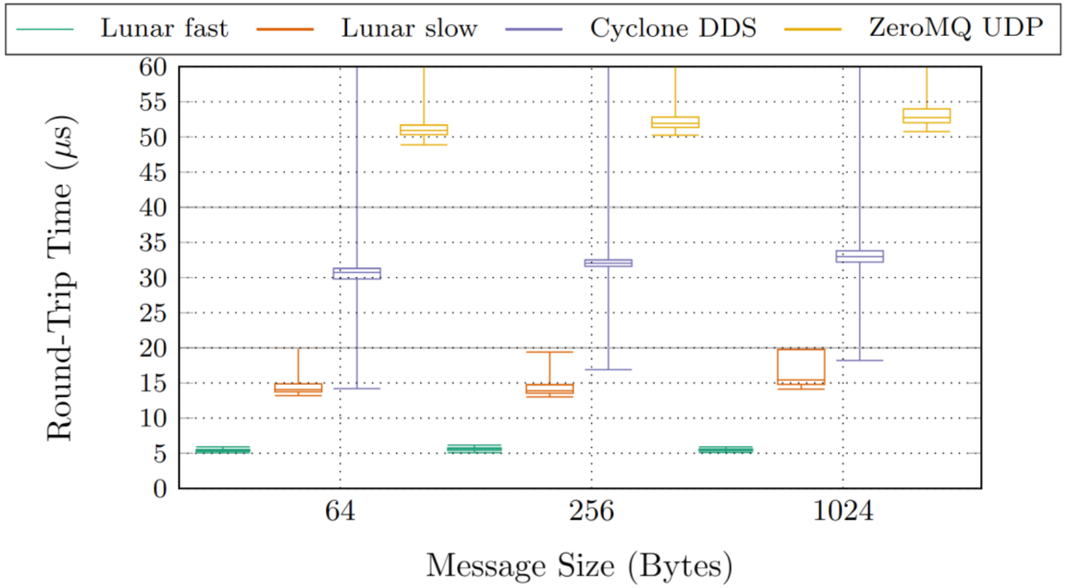
- **OS**: Ubuntu 22.04
- **CPU**: 18-core Intel I9-10980XE @ 3.00Ghz
- **RAM**: 64GB
- **NIC**: Mellanox DX-6 100Gbps

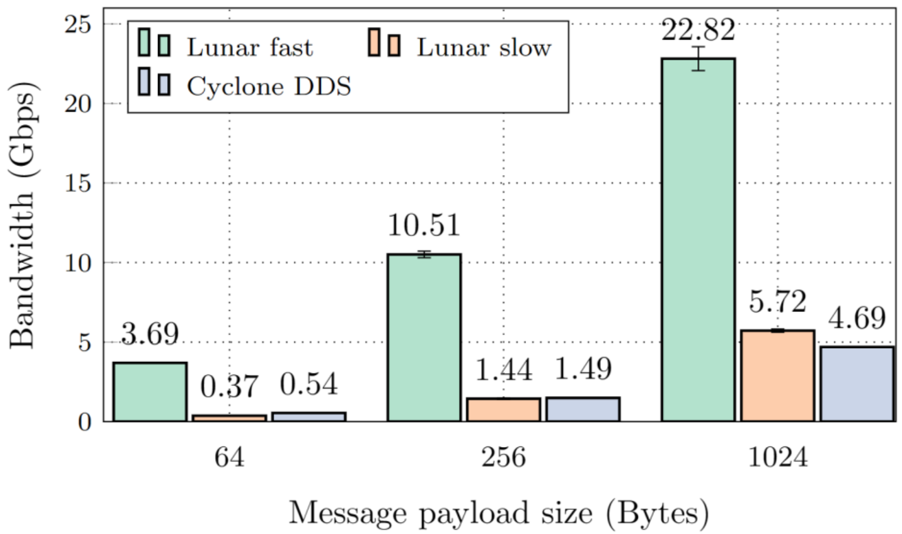
Evaluation running a lunar-based ping-pong application for the RTT, and one publisher to one subscriber for the throughput

Comparison with other State-of-the-Art MOM: **Cyclone DDS [1]** and **ZeroMQ [2]**

In the accelerated case (Lunar fast) Lunar MOM outperforms the other solutions in both RTT and throughput tests.

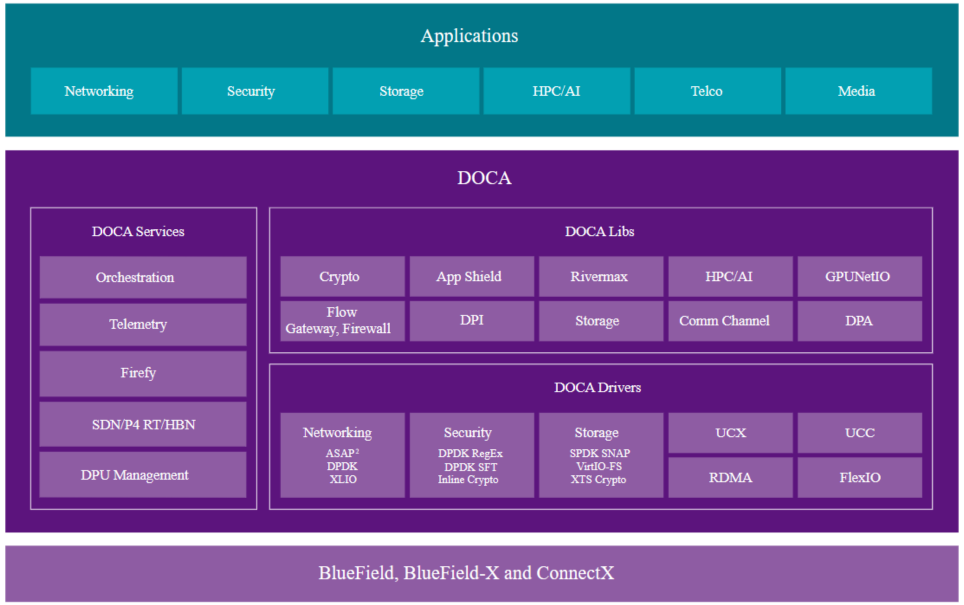
[1] https://cyclonedds.io/
[2] https://zeromq.org/





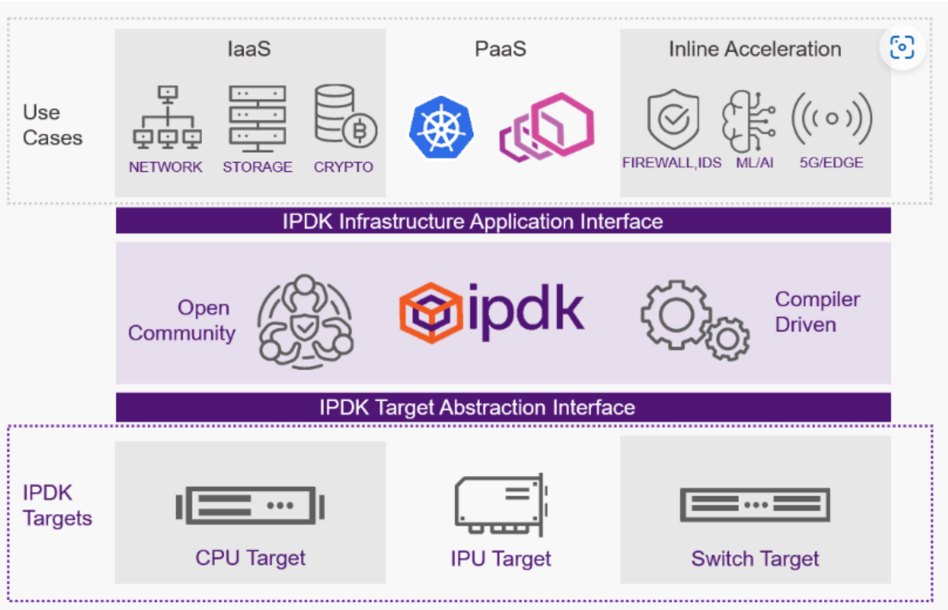
ZeroMQ is excluded as it showed unstable performance during the tests

Future Work (1/2): Are those Approaches Just for the Network? Considering the Infrastructure



NVIDIA DOCA

https://developer.nvidia.com/networking/doca



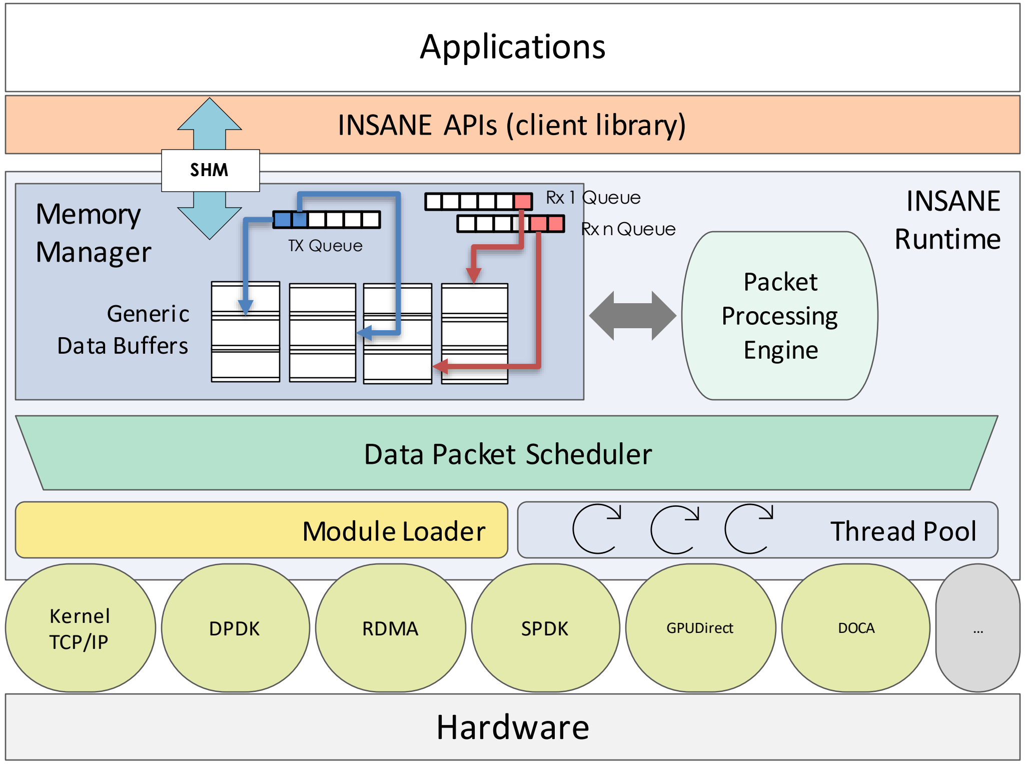
ipdk Infrastructure Programmer Development Kit

https://ipdk.io/

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Future Work (2/2): Extensions towards full programmability

- Architectural enhancements:
    - Transition from **context-specific memory pools** to a pool of **Generic Data Buffers**
    - Incorporate different acceleration technologies as **loadable modules on-demand**.
- Programmability: exploiting compute resources for **in-network processing** and **observability**

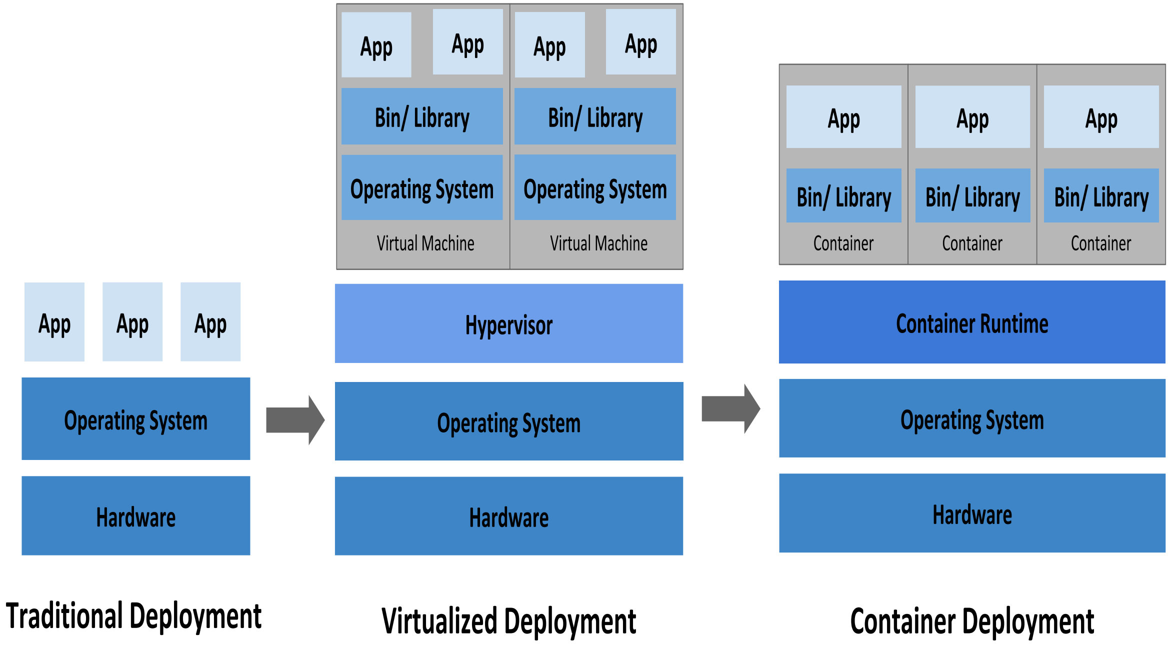


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

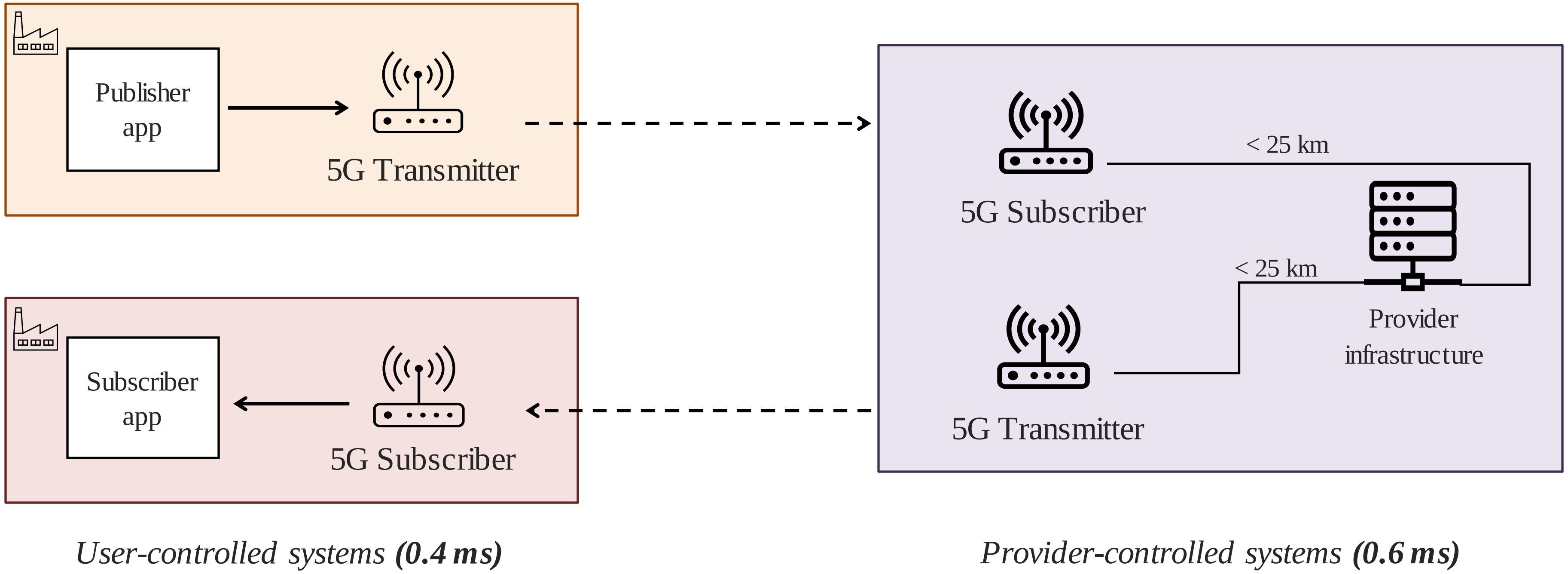
E2E TSN Orchestration
&
TSN-enabled Virtual Environments

Statement: A Cloud Perspective – Everything-as-a-Service

- Cloud computing benefits from **scale economy** of general-purpose technologies offered *as-a-Service*.
- Best suited for *elastic workloads, e.g., big data batches*.
- Paid in terms of **many software layers**, introducing unacceptable **overhead,** especially for URLLC environments.
- Lack of ***as-a-Service*** offers **supporting URLLC** solutions in phy/virtualized (commodity) **soft-real time environments**.



Ultra-Low Latency 5G applications in VM-based Environments



ULL applications require sub-millisecond end-to-end latency. Most of this ***budget*** must be allocated for ***external provider operations***: *end-host processing must be extremely efficient* [1]

To fulfil these constraints, application rely on ***networking technology that are at odds with virtualization***

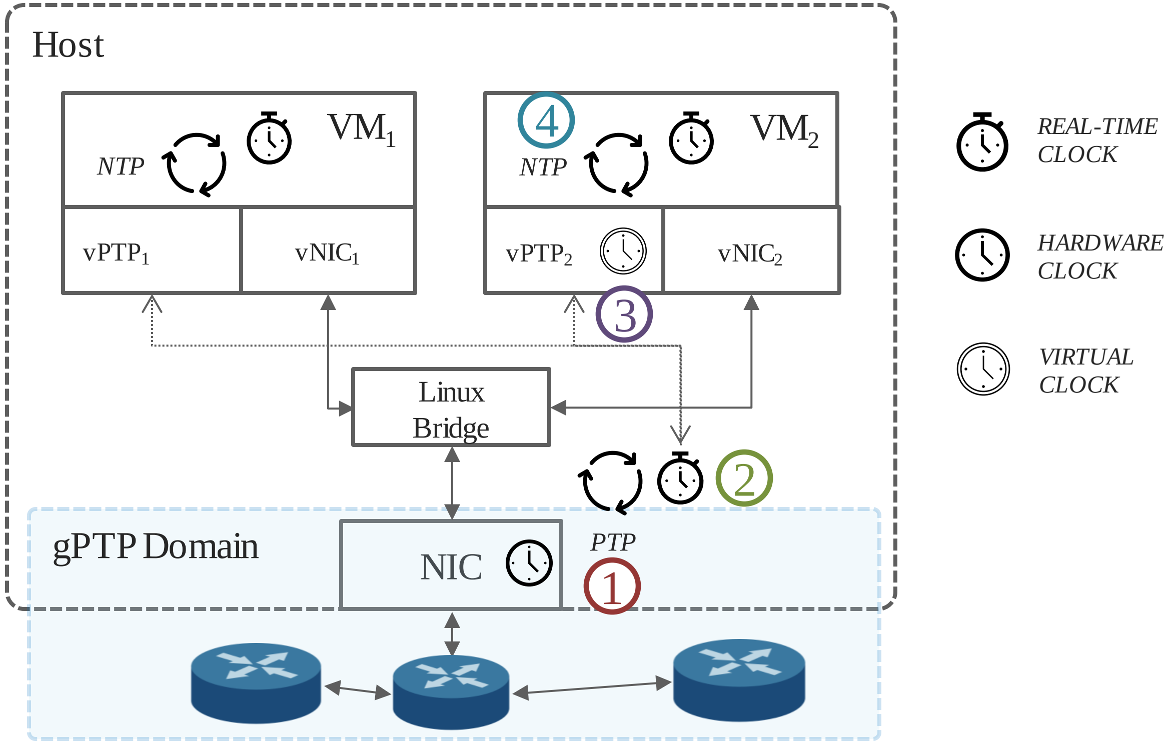
[1] Xiang, Zuo, et al. "Reducing latency in virtual machines: Enabling tactile Internet for human-machine co-working." IEEE Journal on Selected Areas in Communications 37.5 (2019): 1098-1116.

TSN-enabled Virtual Environments for URLLC: Virtual PTP clocks

To run unmodified TSN applications in ***VM***, we provide each VM with a **virtualized clock** that tracks the host's ***system clock***

SYNCHRONIZATION STEPS

1. ***Host NIC*** sync with the rest of the ***network via PTP***
2. The PTP process synchronizes the ***NIC clock*** with the ***host system clock***
3. ***VMs virtual clocks sync with host clocks***
4. A ***Network Time Protocol*** (NTP) process synchronize the ***VM's system clock*** using the virtual clock as a reference

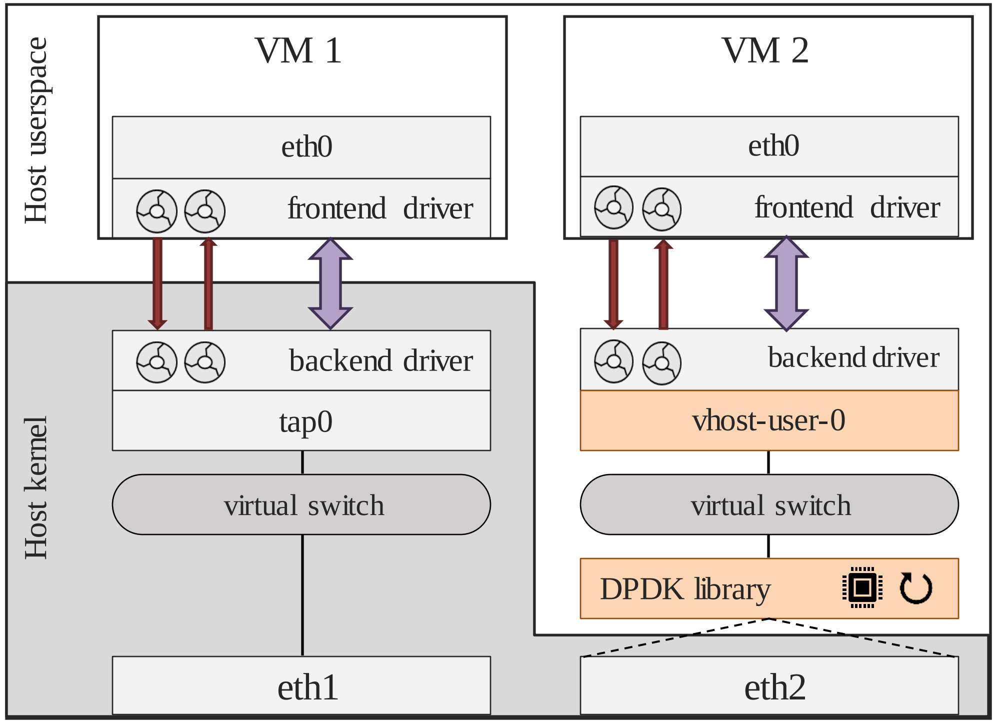


ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

TSN-enabled Virtual Environments for URLLC: Optimizing the Datapath

Main sources of **datapath overhead** are in the kernel networking stack: *data copies, context switches, etc*. [1]

However, ***TSN scheduling*** is performed by the ***guest networking stack***



Kernel-based paravirtualization

Userspace paravirtualization

OUR SOLUTION

- ***Guest kernel*** is untouched
- ***Host kernel*** is bypassed using **DPDK**, a library for userspace packet processing.

[1] Cai, Qizhe, et al. "Understanding host network stack overheads." Proceedings of the 2021 ACM SIGCOMM 2021 Conference. 2021.

TSN-enabled Virtual Environments for URLLC: Evaluation Setup

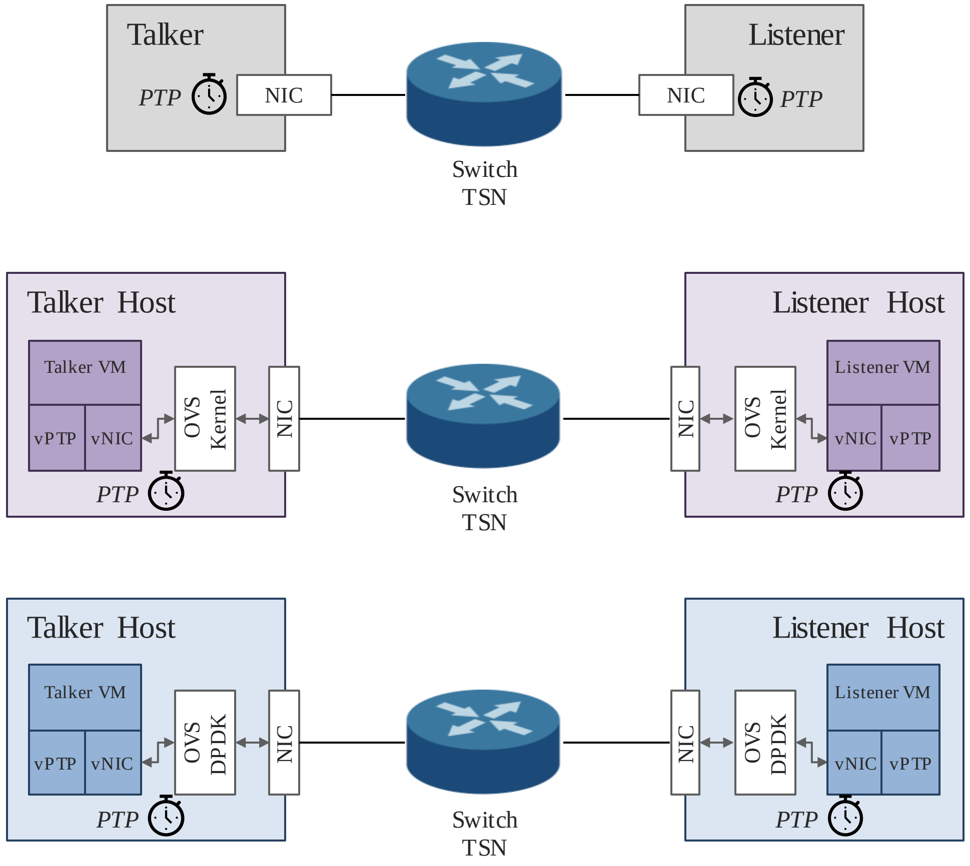
GOAL Show that ***virtualized TSN applications*** preserve ***sub-millisecond E2E latency*** and good ***determinism***.

Assuming ***60%*** of the budget reserved for ***5G WAN propagation,*** we set the ***latency threshold to 0.4 ms***

TESTBED SETUP

- ***2 UP Xtreme boards*** (***Talker*** and ***Listener***)
  - 4 1Gbit TSN NICs (Intel I210)
  - Intel Core i3-8145UE CPU with 2/4 cores
  - 8GB of RAM
  - OS - Ubuntu 20.04 with Linux kernel 5.4.0
- ***1 TSN-compliant*** Relyum RELY-TSN-BRIDGE ***switch***

TEST APPLICATION One ***publisher***, one ***subscriber***, each running ***baremetal*** or in ***VMs*** on a separate hosts. Exchange UDP packets with ***1ms publishing cycle***

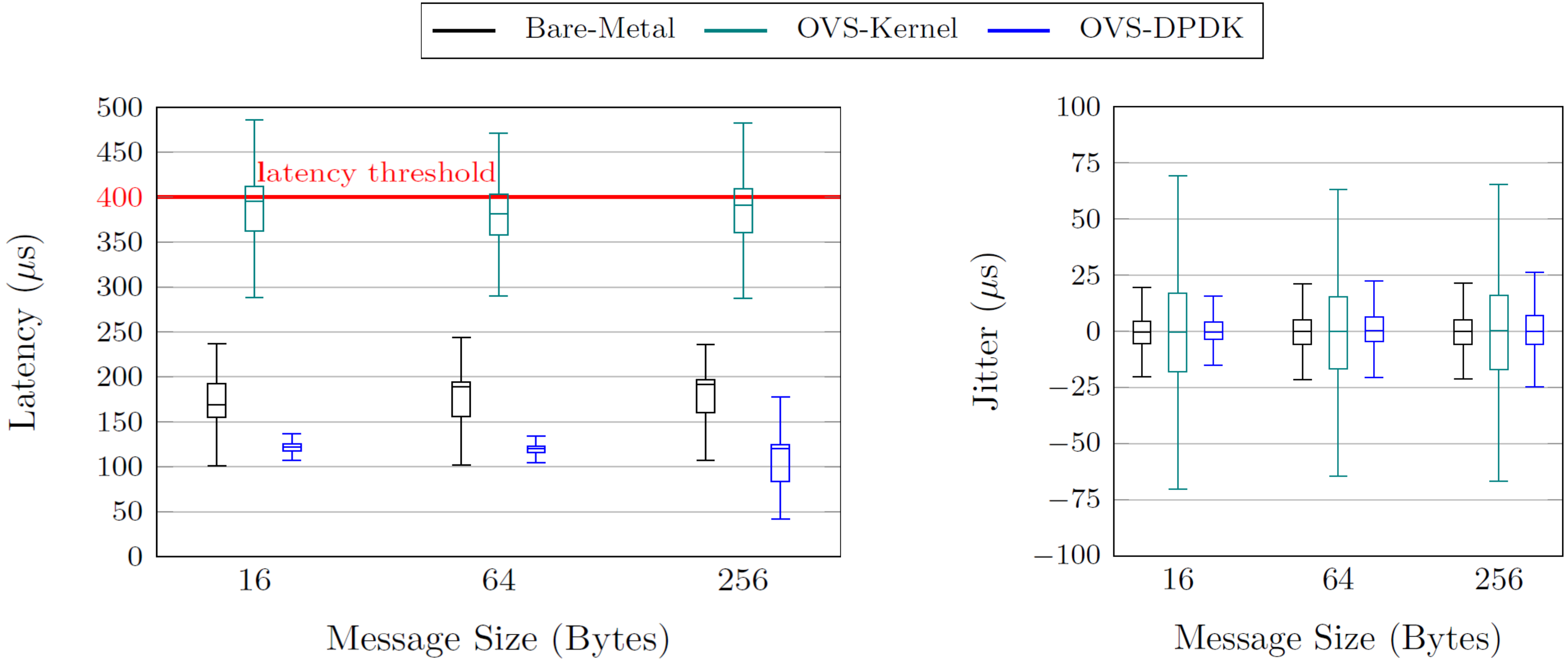


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

TSN-enabled Virtual Environments for URLLC: E2E Latency and Jitter Evaluation

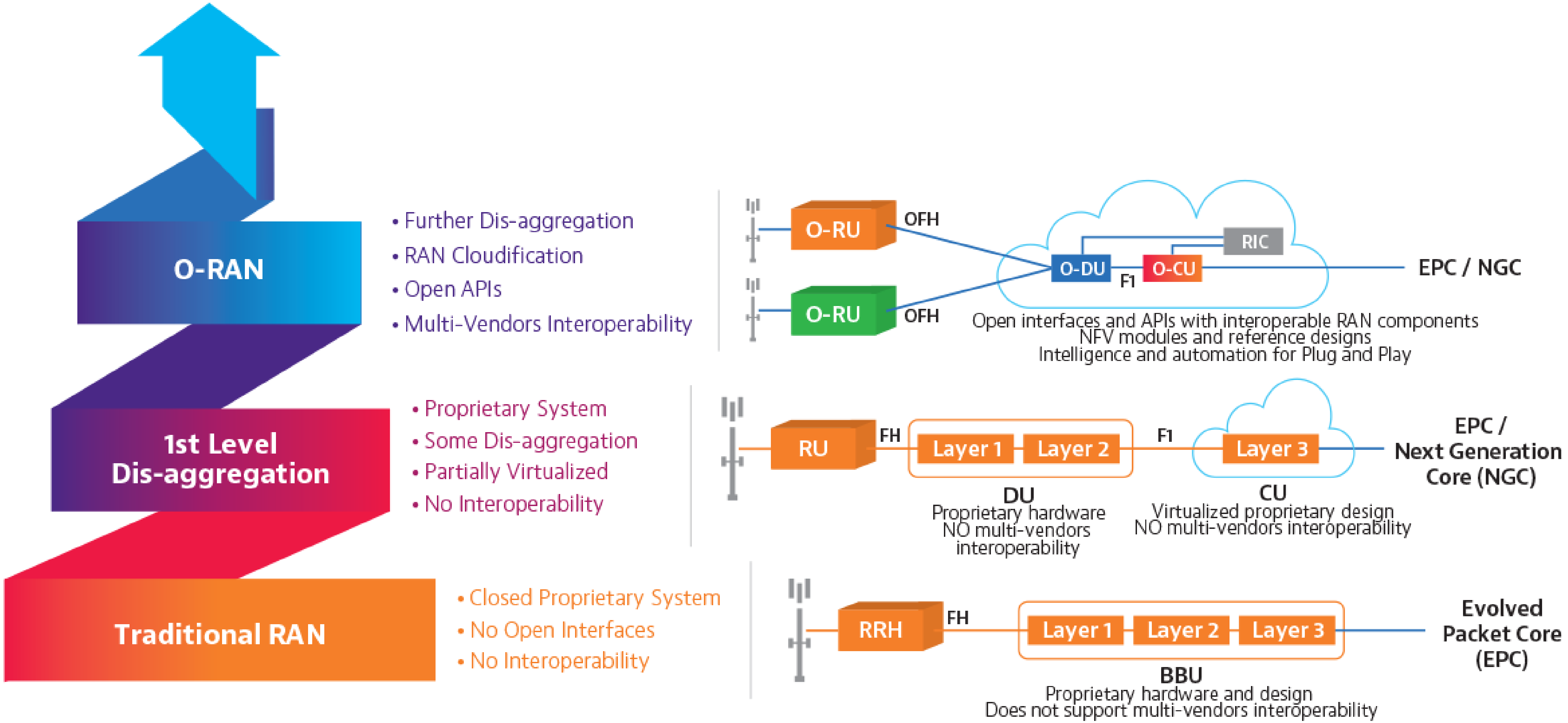
The experiment demonstrates our solution's ability to ***support TSN applications in virtual environments***, with latency stable **< 400us**

→ The ***OVS-DPDK setup*** performs even better than bare-metal thanks to kernel bypassing



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

TSN-enabled Virtual Environments for URLLC: 5G RAN Disaggregation



Resource disaggregation and containerization in the C2TC

Typical C2TC applications are ***disaggregated*** and ***containerized***:

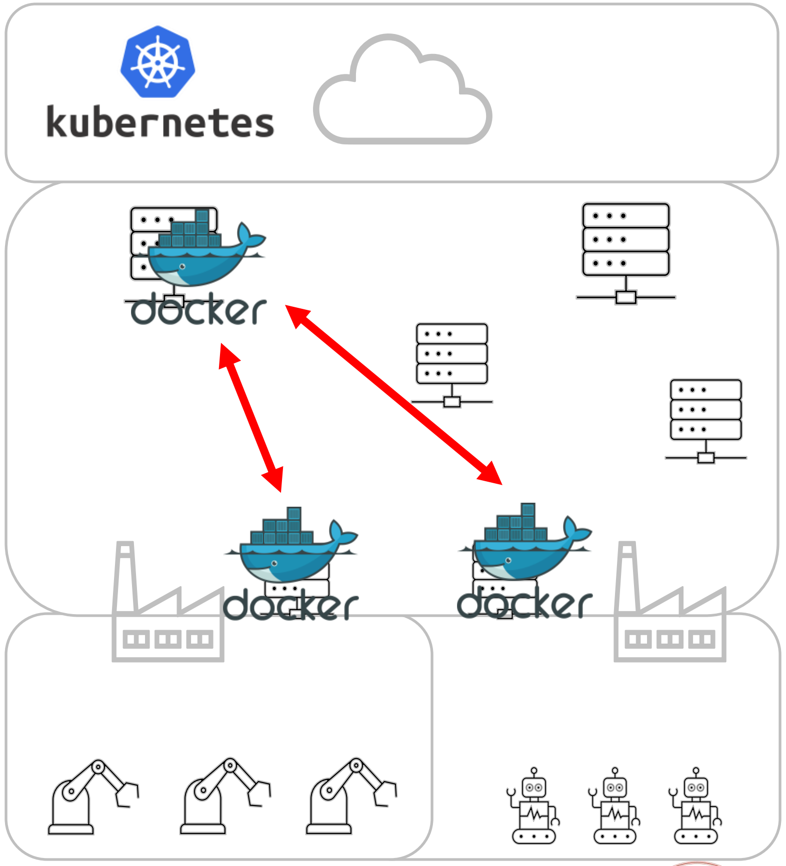
- Applications are designed as ***interacting components***
- Each component lives in an ***isolated container***

Orchestrators (e.g., Kubernetes) can optimize the placement of these components based on:

- Application requirements
- Edge node capabilities (e.g., CPU available)

However, orchestrators currently ***do NOT consider networking*** in their placement decision, even if network delays might ***disrupt the operations*** of application requiring:

1. Ultra-Low Latency
2. Deterministic behavior



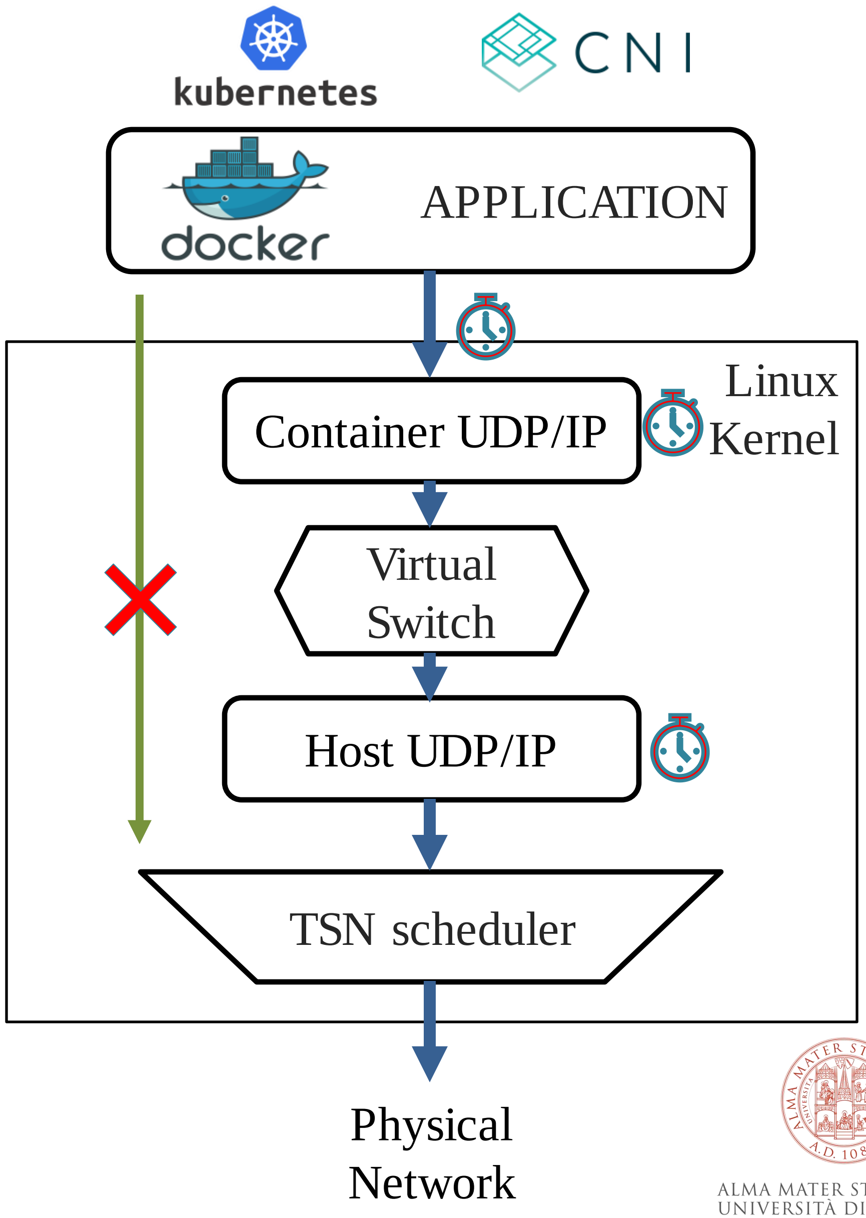
Flannel and the overhead of container overlay network

Kubernetes can create ***overlay networks*** among containers through a set of ***Container Network Interface (CNI)*** plugins

There are many CNI plugins, e.g., *Flannel*

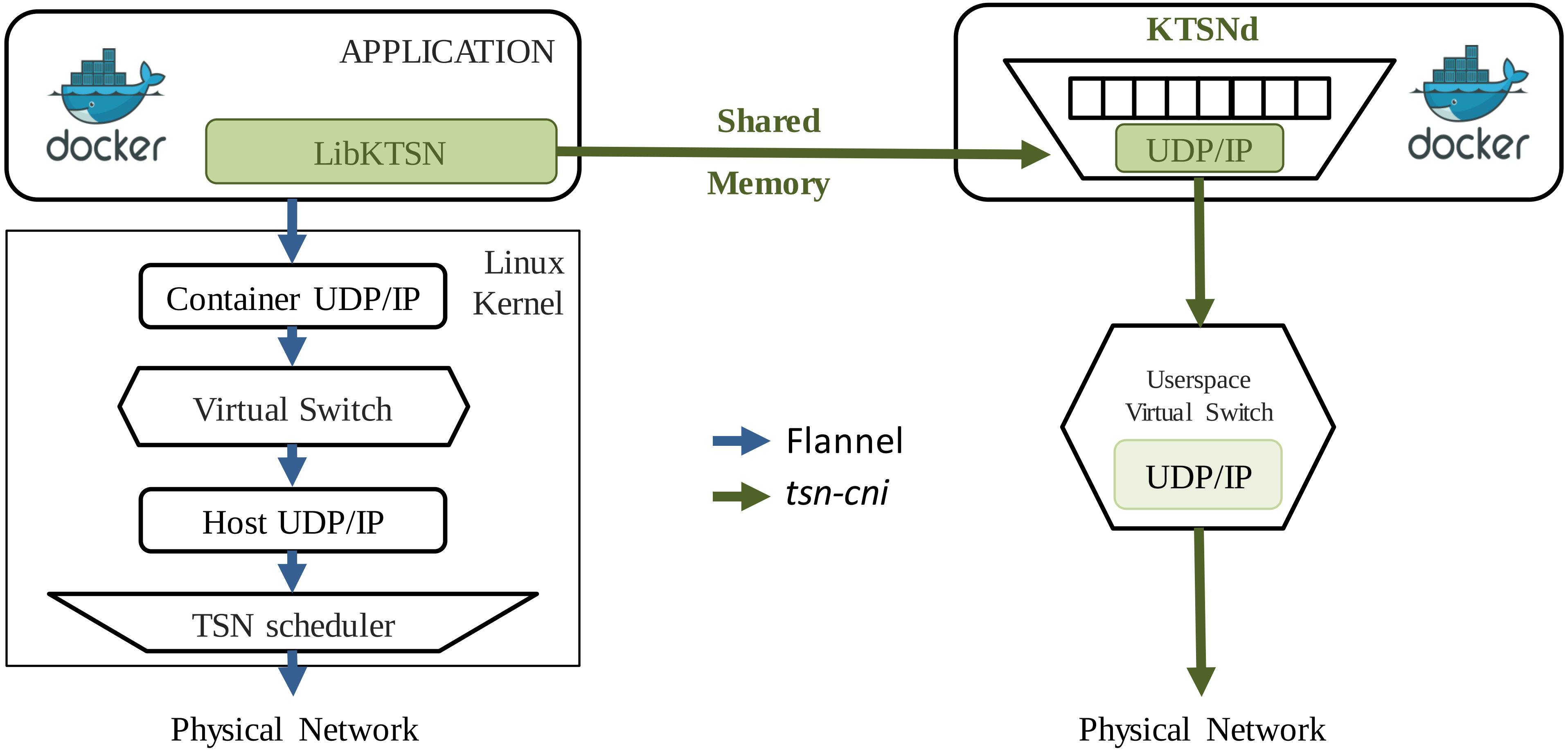
Although with different tools and mechanisms, they set up the same network configuration, introducing ***two crucial issues*** for mission-critical applications:

1. Multiple instances of the ***kernel-based*** network stack
2. Impossibility to ***configure TSN*** protocol parameters



KuberneTSN: A CNI for Time-Sensitive Applications in the Computing Continuum

KuberneTSN is a ***new Kubernetes CNI*** that allows ***deterministic communication*** between containers through a ***new userspace TSN scheduler*** and achieves ***ULL*** through a ***kernel-bypassing, zero-copy datapath***



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

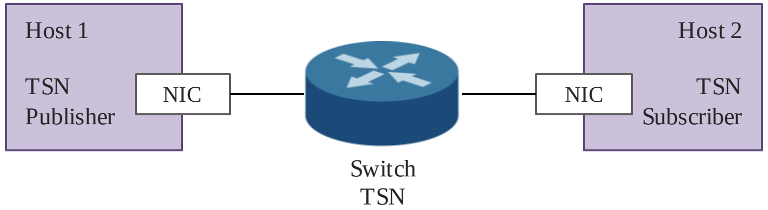
KuberneTSN: Evaluation Setup

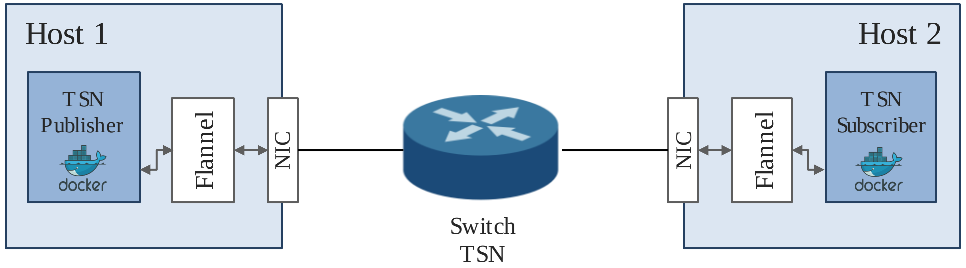
GOAL Show that ***tsn-cni*** can configure the network to achieve ***low-latency*** and a ***deterministic behavior***

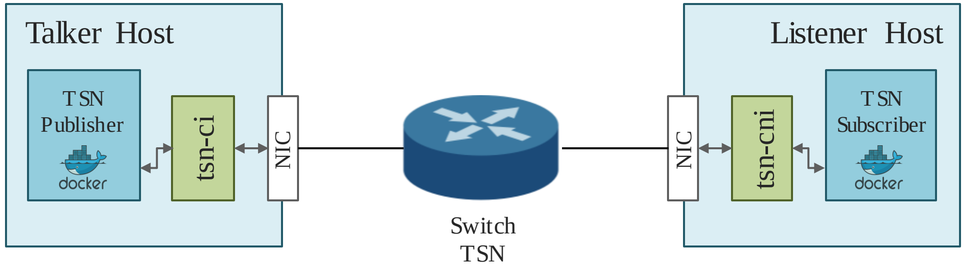
TESTBED SETUP

- 2 machines equipped with
  - 2.5Gbit Intel I225 NIC
  - Intel i9-10980XE 18/36 CPU
  - 64GB RAM
- 1 TSN-compliant switch

TEST APPLICATION ***1 pub***, ***1 sub***, running in containers on separate hosts, exchange ***UDP packets*** with ***1ms publishing cycle***



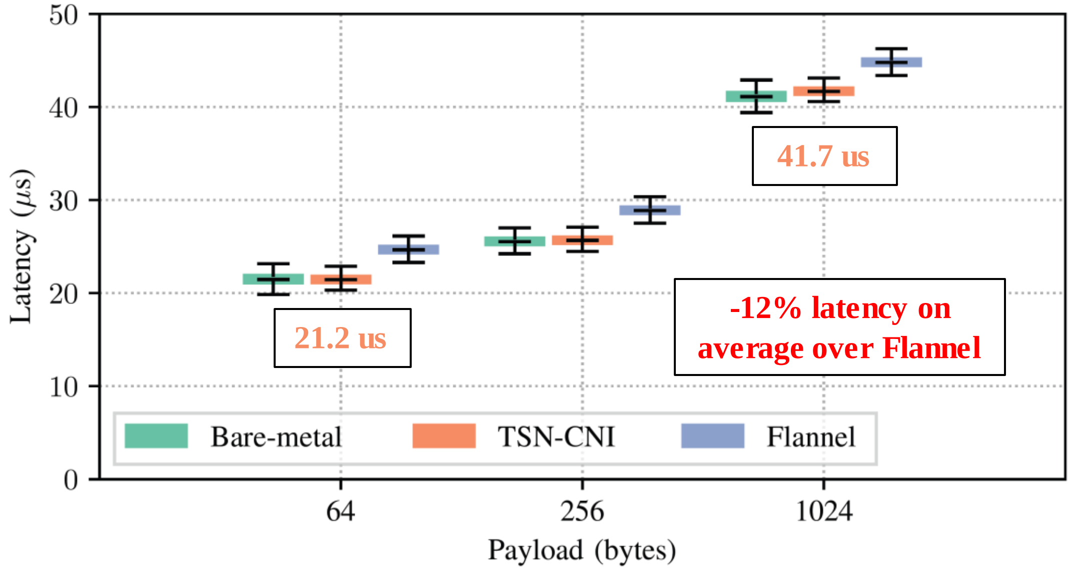


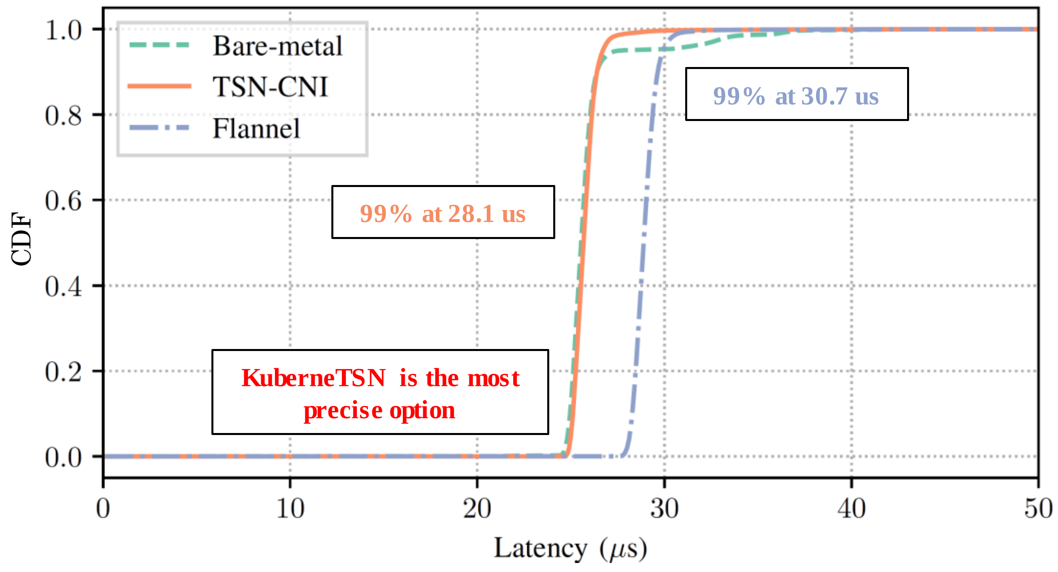


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

KuberneTSN: End-to-End Latency & Determinism Evaluation

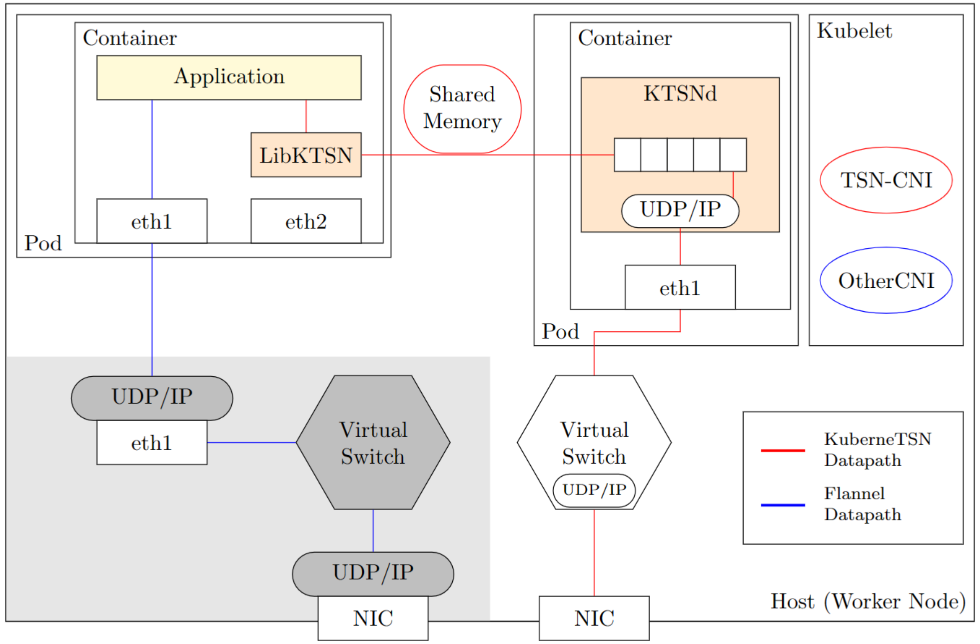
We consider ***different data sizes*** (64, 256, 1024 bytes) and compare the results to a ***bare-metal deployment*** and a ***typical Flannel configuration***





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

KuberneTSN: Future Work

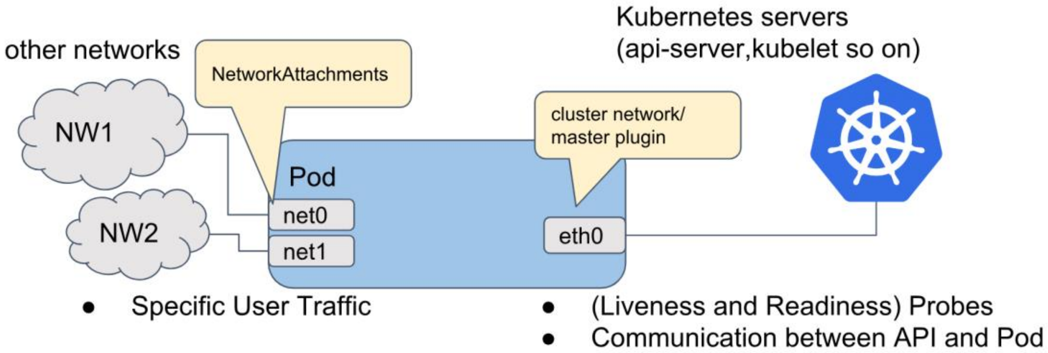


- Kubernetes can create **overlay networks** among containers through a set of *Container Network Interface (CNI)* plugins

- Currently ***do not consider networking*** for placement decision, even if network delays might disrupt the operations of application requiring: *ULL, deterministic behavior*

- **PROBLEM**: two **crucial issues for mission-critical applications**
  - Multiple instances of the kernel-based network stack
  - Impossibility to configure TSN protocol parameters

- **POSSIBLE SOLUTION**:
  - KuberneTSN defines a **new Kubernetes CNI plugin**: kernel-bypassing zero-copy datapath, userspace TSN scheduler
  - Integration of **Multus CNI** → Kubernetes plugin that enables attaching multiple network interfaces to pods.
  - **Layered orchestrator for network/compute resources**



[1] A. Garbugli, L. Rosa, A. Bujari and L. Foschini, "KuberneTSN: a Deterministic Overlay Network for Time-Sensitive Containerized Environments," ICC 2023 - IEEE International Conference on Communications, Rome, Italy, 2023, pp. 1494-1499, doi: 10.1109/ICC45041.2023.10279214.

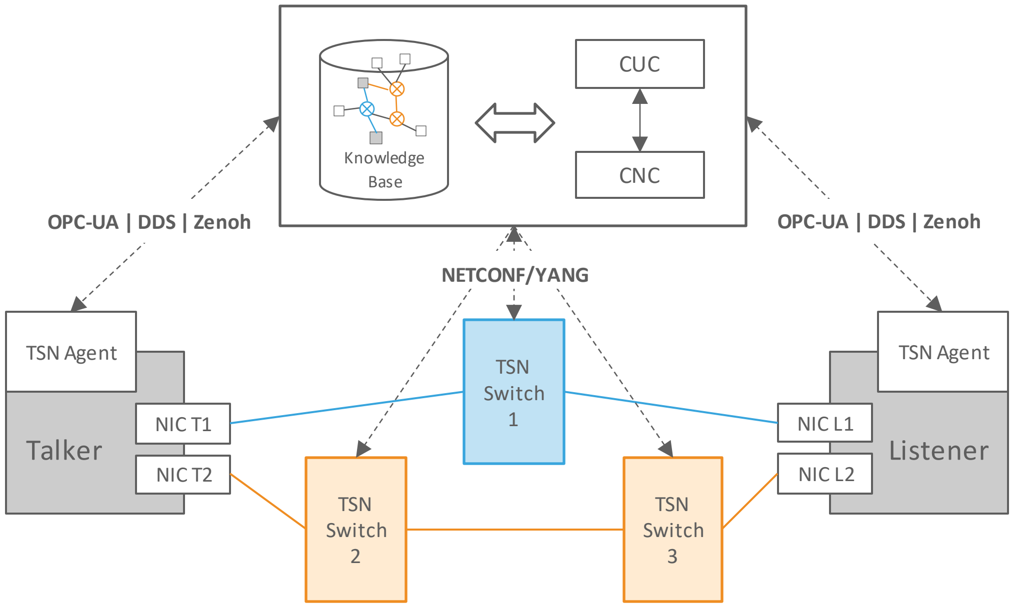
E2E TSN Orchestration: QoS-Aware Management and Control

The ***Centralized Network Configuration*** (CNC): Manages the networked devices enforcing QoS as requested by TSN communication endpoint: ***time synchronization***, ***VLAN setup***, and ***network schedule***

The ***Centralized User Configuration*** (CUC): Cooperate with the TSN agent to manage the ***TSN stream*** required by the end devices

The ***Knowledge Base***: ***Stores information*** regarding managed elements. Information can be sent by the participants or requested directly from the knowledge base.

The ***TSN Agent***: Query the CUC to request valid QoS-aware TSN flows and uses ***Netlink*** as a protocol to manage and monitor the status of a TSN stream



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

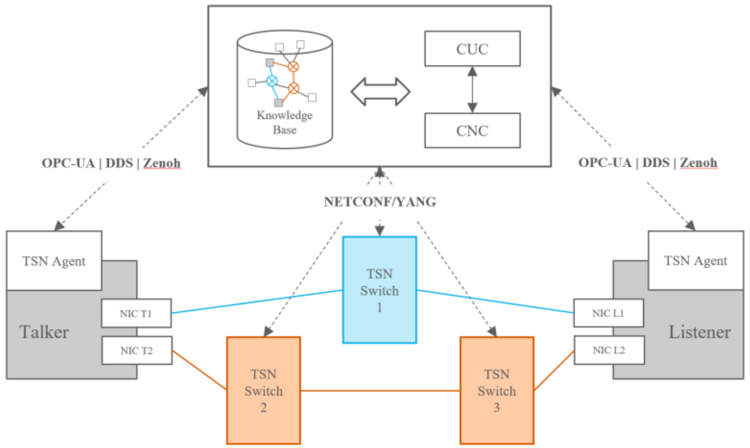
E2E TSN Orchestration: Latency and Jitter Evaluation

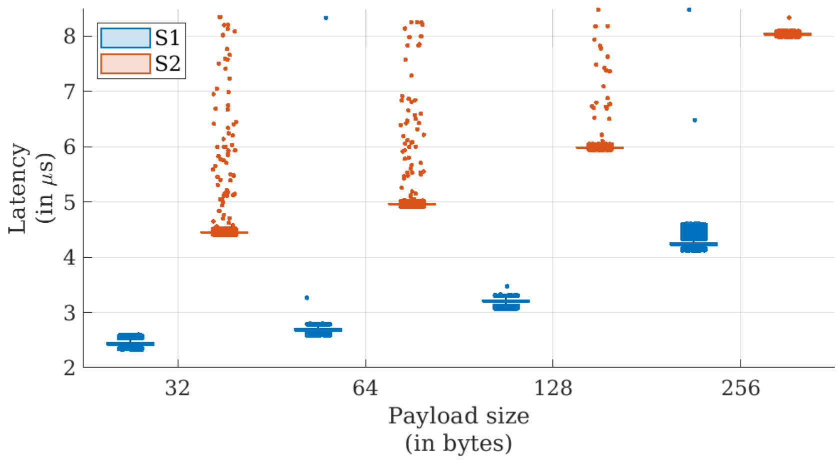
We used a ***UDP-based*** traffic with a payload varying from ***32, 64, 128 and 256 bytes***, and with a ***regular interval of 1 ms***

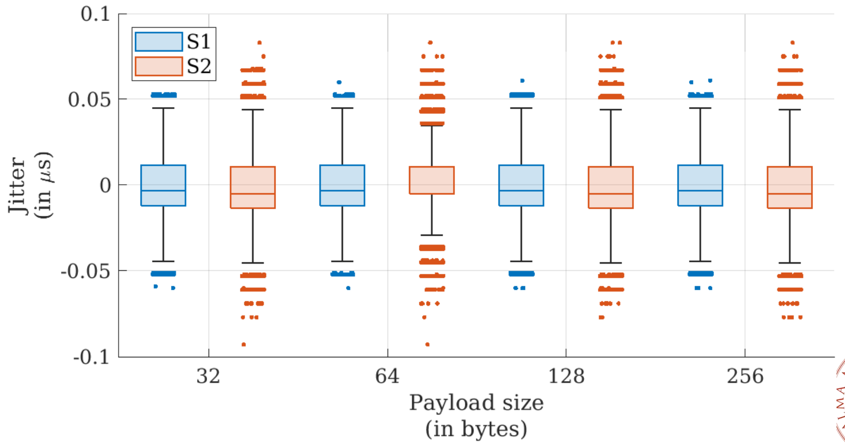
TESTBED SETUP

- ***3 UP Xtreme boards*** (***Talker***, ***Listener***, and ***CUC+CNC+Knowledge Base***)
  - 4 1Gbit TSN NICs (Intel I210)
  - Intel Core i3-8145UE CPU with 2/4 cores
  - 8GB of RAM
  - OS - Ubuntu 20.04 with Linux kernel 5.4.0

- ***3 TSN-compliant*** Relyum RELY-TSN-BRIDGE ***switches***
  - Each switch has 4 1Gbit ports





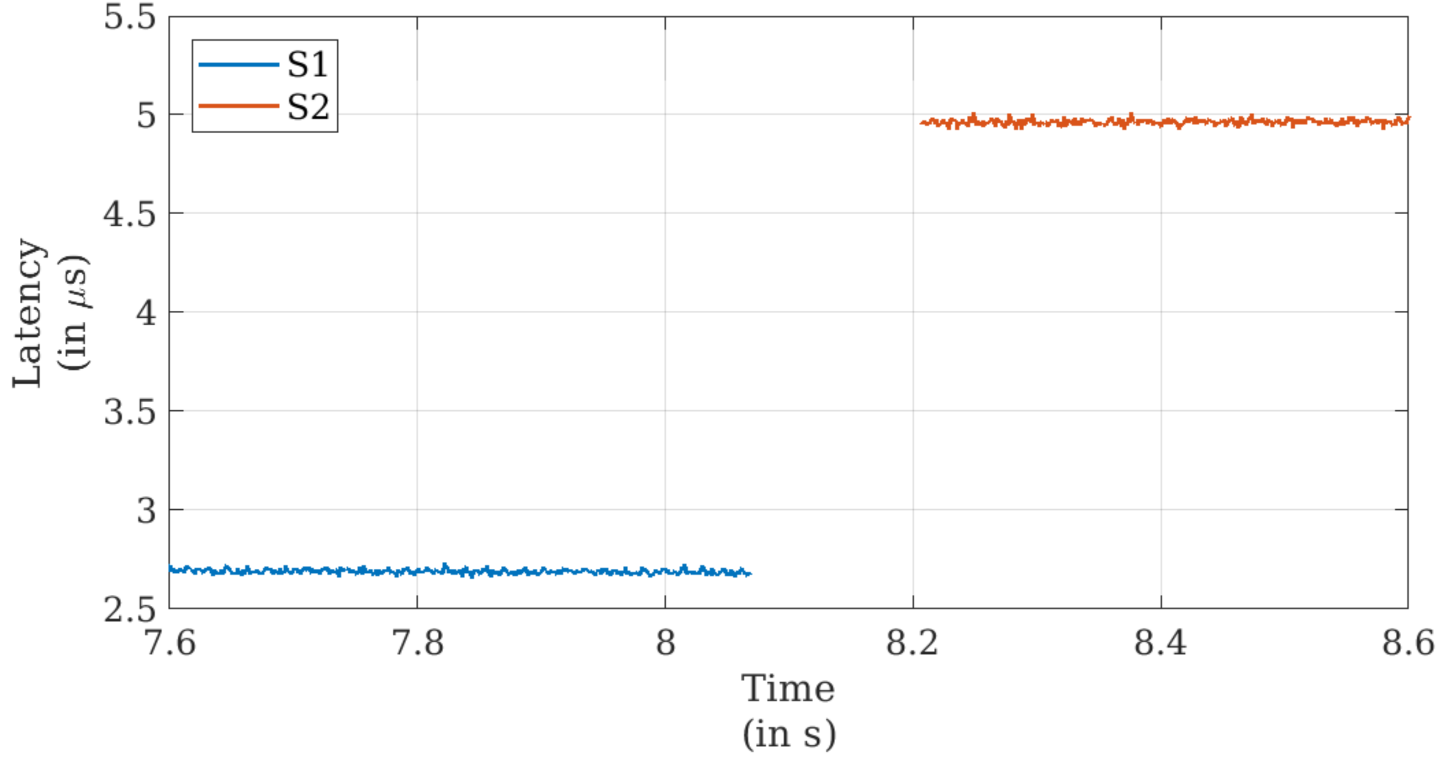


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

E2E TSN Orchestration: Reconfiguration Scenario

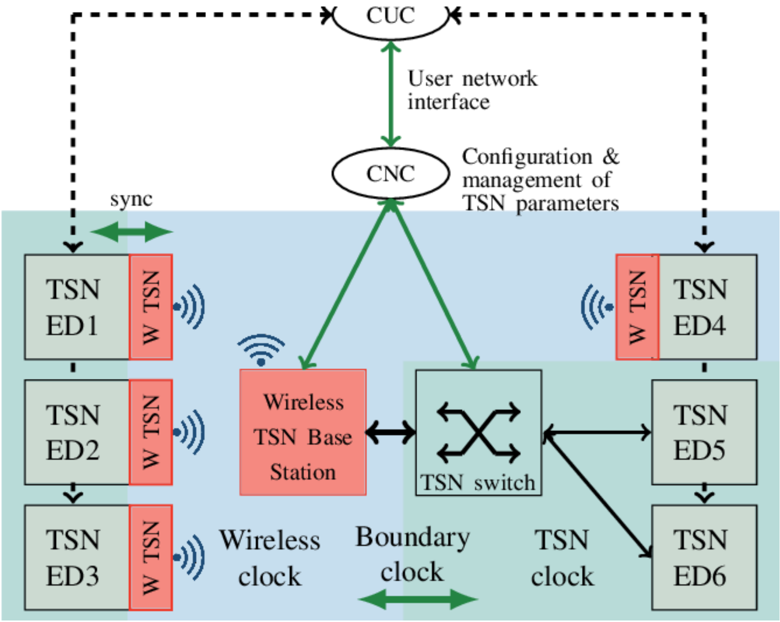
The experiment demonstrates the ability of TSN agents to handle ***reconfiguration events***.

→ The communication is initially serviced via the ***S1*** and, following a link-drop event, goes through ***S2*** with a downtime period due to the reconfiguration of about **150 ms**

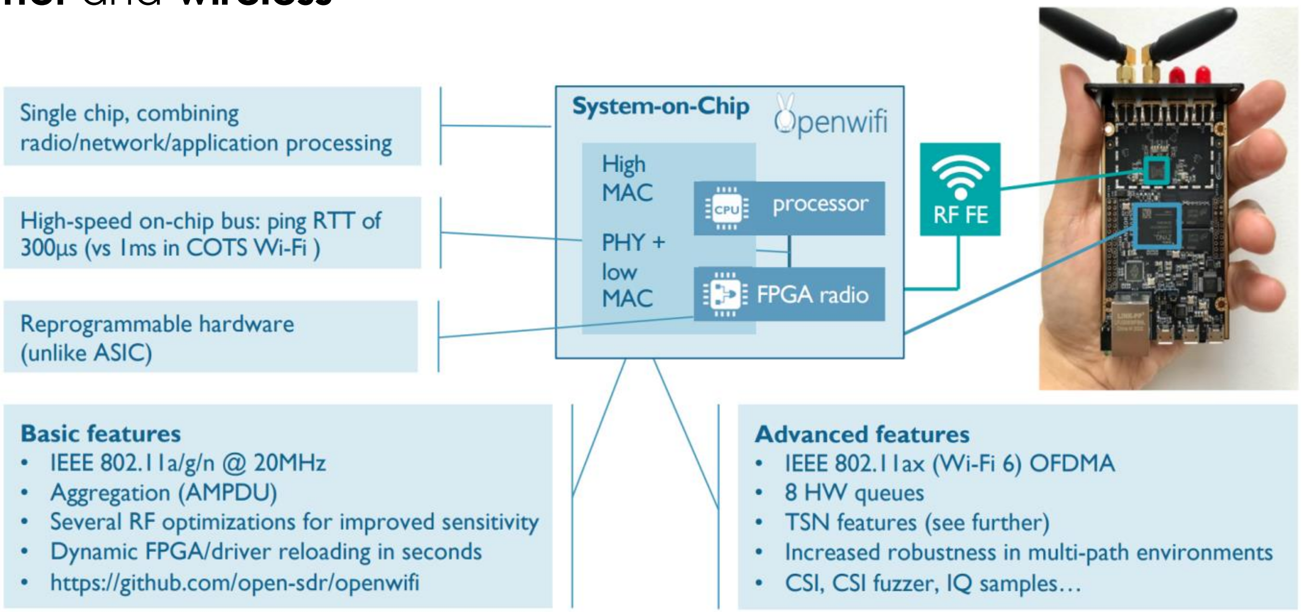


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Future Work: Extending with Wi-Fi



- Increased **Mobility in Industrial Automation**: Reducing wiring complexity, enabling **versatile network topologies**
- **Applications in Robotics and AGVs**: Supporting real-time applications in automated vehicles and industrial robotics
- Integrating **deterministic networking over Ethernet** and **wireless** infrastructure
- **Wi-Fi Integration for Wireless TSN**: Increasing network range and flexibility, addressing latency and jitter challenges
- **SOLUTION**: Utilizing an **open-source** Wi-Fi stack for adaptable and customizable TSN solutions, e.g., **openwifi** [1]



[1] open-sdr/openwifi: open-source IEEE 802.11 WiFi baseband FPGA (chip) design: driver, software (github.com)

Concluding Remarks: The Future of Industrial Networking

Integration of TSN in Industrial Networks → TSN is revolutionizing industrial network communications, ensuring high reliability, and deterministic data delivery

Synergy with Emerging Technologies → TSN, in **conjunction** with advancements in **5G**, **Wi-Fi 7**, and **cutting-edge hardware solutions**, can lead to more efficient, flexible, and scalable industrial networks

Future Directions → Anticipate further **integration of TSN with emerging wireless technologies**, enhancing the capabilities of Industrial IoT, Smart Cities, and Automated Vehicles. The move towards a **data-centric network architecture** and **specialized hardware acceleration** opens new possibilities for innovation and efficiency in industrial applications

Challenges and Opportunities → Challenges in standardizing and deploying these technologies on a large scale. However, significant opportunities for research, development, and innovation in the field of industrial networking

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Andrea Garbugli

Armir Bujari

Department of Computer Science and Engineering (DISI)

University of Bologna – Italy

andrea.garbugli@unibo.it

armir.bujari@unibo.it

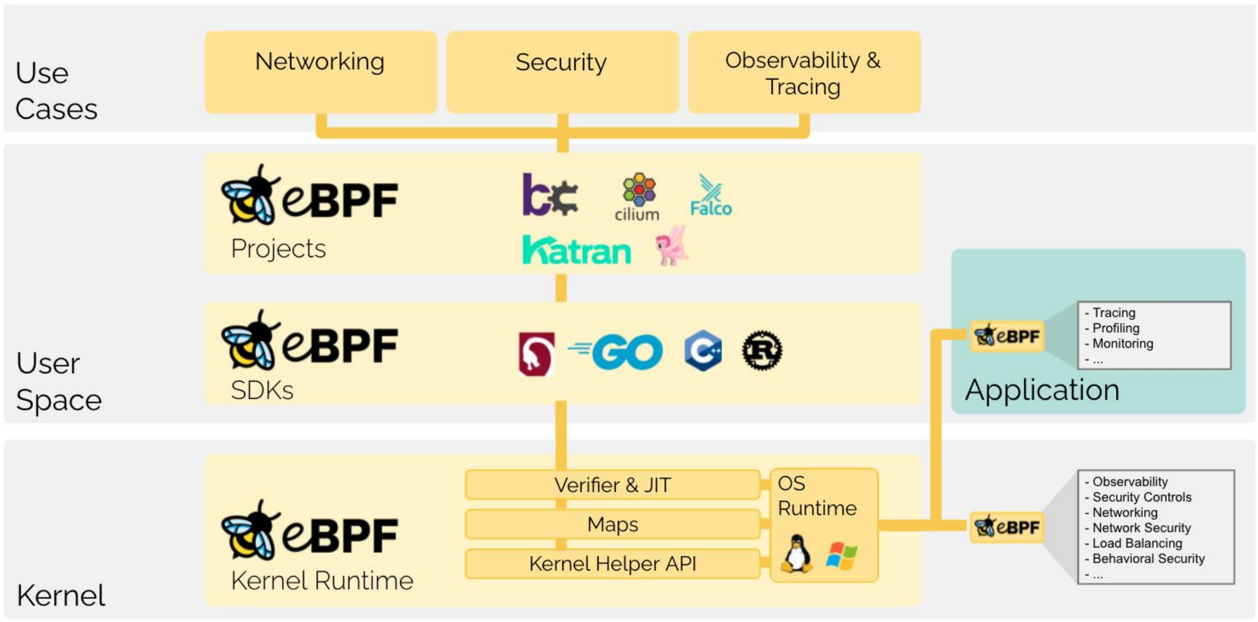
Introduction to eBPF Technology

- The operating system kernel is crucial for implementing observability, security, and networking due to its comprehensive system control.

- However, kernel evolution is challenging due to its central role and the need for high stability and security

  → **eBPF**, is a technology that allows running sandboxed programs in privileged contexts like the Linux kernel (now also in Windows)

- **eBPF** enables extending kernel capabilities without altering kernel source code or loading kernel modules

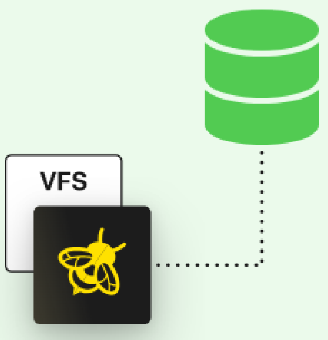


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

What's possible with eBPF?

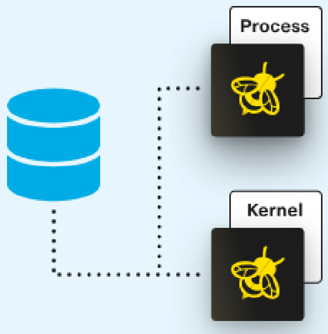
Networking

Speed packet processing without leaving kernel space. Add additional protocol parsers and easily program any forwarding logic to meet changing requirements.



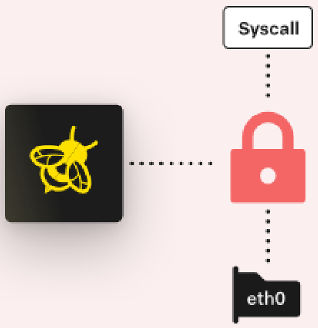
Observability

Collection and in-kernel aggregation of custom metrics with generation of visibility events and data structures from a wide range of possible sources without having to export samples.



Tracing & Profiling

Attach eBPF programs to trace points as well as kernel and user application probe points giving powerful introspection abilities and unique insights to troubleshoot system performance problems.



Security

Combine seeing and understanding all system calls with a packet and socket-level view of all networking to create security systems operating on more context with a better level of control.