FUTUREWEI
Technologies

Enabling Technologies for Operations and Control Networks - Tutorial

K. Makhijani, C. Westphal, L. Dong

IEEE NOMS 2023
12th May, Miami, FL, USA

Team

Cedric Westphal
Principal Research Architect
Cedric.westphal@Futurewei.com

Lijun Dong
Principal Architect
Lijun.dong@Futurewei.com

Kiran Makhijani
Principal, Standards and Research
kiranm@Futurewei.com

Part –I/III

Formalize OCN Concept and its Description

Foundations of the OCN Model

Part -II

Conceptual notion of the OCN

Use cases Deep dive to rationalize Motivation and Requirements

Part -III

Realization of the OCN

Enabling Network Technologies (standards focused)

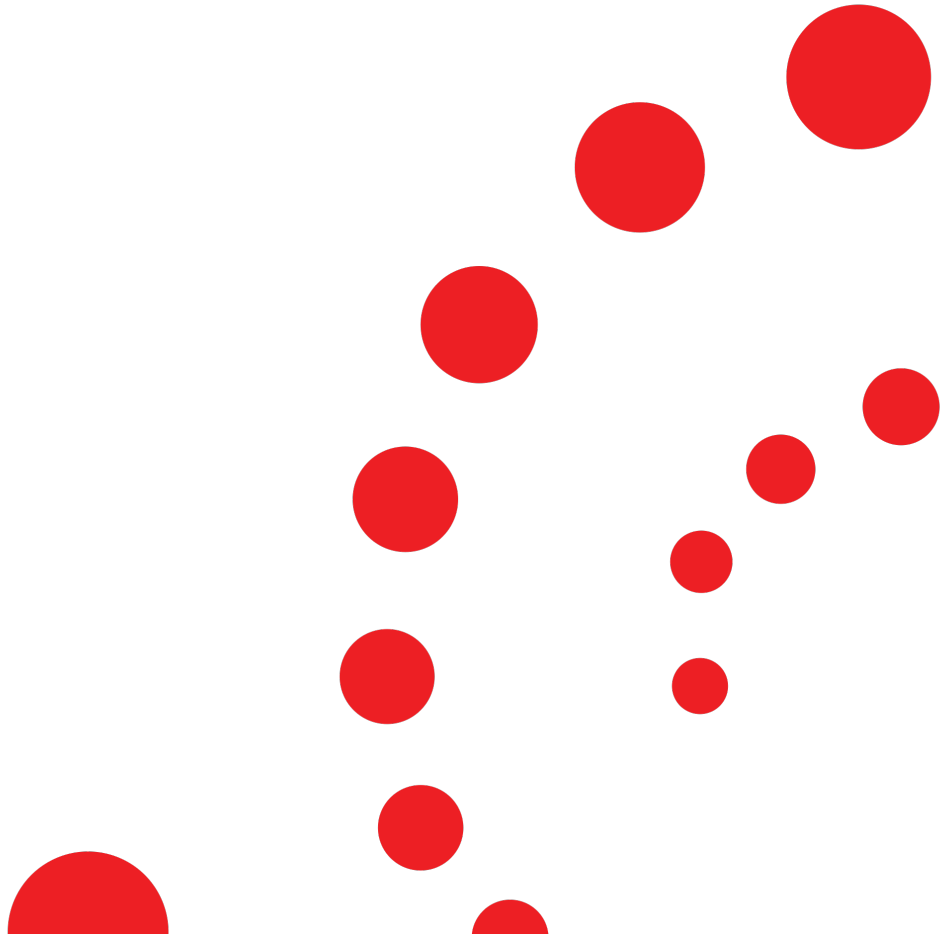
Demo - Snippet

Agenda

- Welcome – Kiran (2 mins)
- Part I - Overview - Cedric (15 mins)
  - Abstract Operations and Control Networks concept
  - Industrial IoT & Factory automation
- Part II - Deep Dive and Detail Description into use cases - Lijun (45mn)
  - Virtual PLC
  - Remote automotive driving, etc.
- Part III - OCN Model - Cedric (25 mins)
  - Generalization and Description
- Coffee Break (15:30 – 16:00 pm)
- Part III - OCN Model –(contd. 20 mins)
  - Components and Communication primitives
- Part IV - OCN Realization - Kiran (60 mins)
  - Enabling Protocols and Emerging Technologies
  - OCN Stack and Demo Part
- Wrap-up

FUTUREWEI
Technologies

Part I: Intro



Outline of this Part

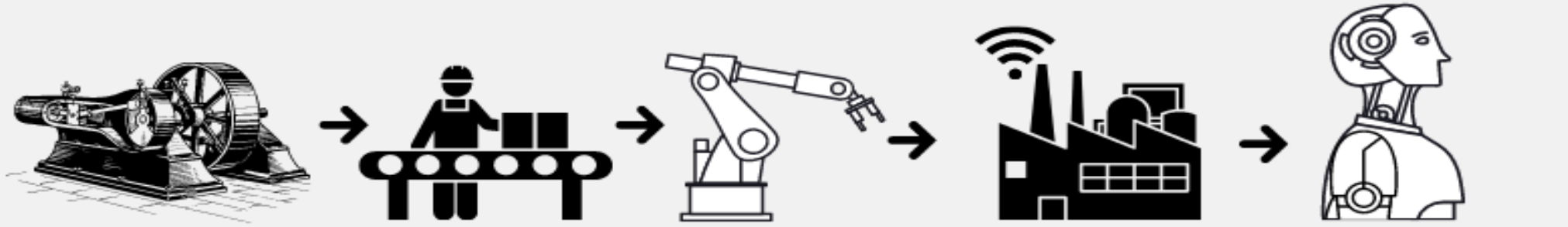
- Brief History of Industrial Networking
- Background on Industrial Networks
  - Current Organization
  - Current Elements & Protocols
- One Use Case: Factory Automation
  - Shortcomings of the current approach
    - Siloed
    - Wall gardens
    - Rigid and lack of feature velocity

Indust

- Industry
- Industry
- Industry
- Industry
- Industry

KNOWHOW

Industrial REVOLUTIONS

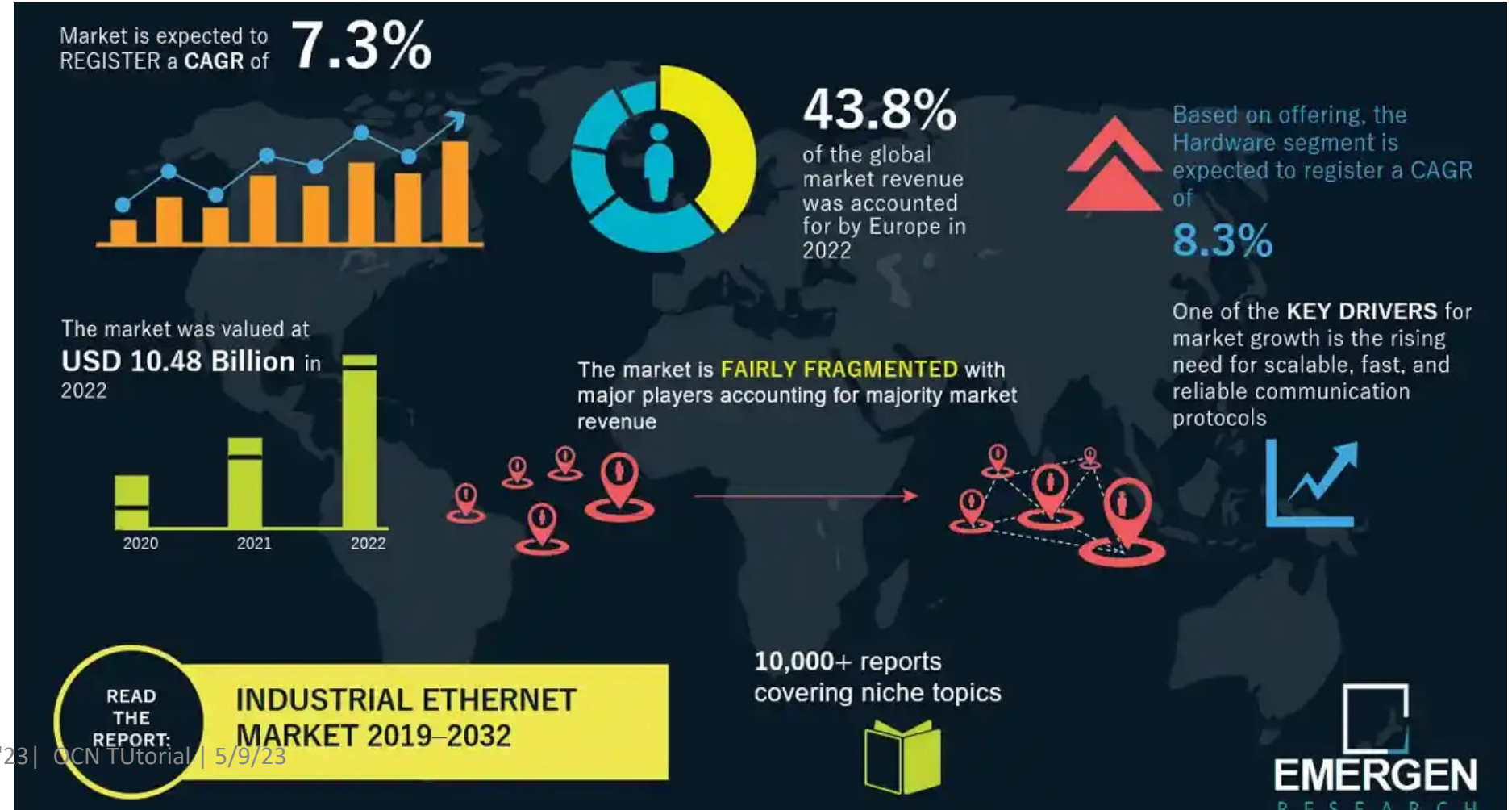


Industry 1.0	Industry 2.0	Industry 3.0	Industry 4.0	Industry 5.0
mechanization, water and steam powers	mass production, electric power, assembly line	computers, automated production, electronics	cyber-physical systems, IoT, networking, machine learning	human-robot collaboration, cognitive systems, customization
1800	1900	2000	2010	2020

om/

Industrial Networking Background

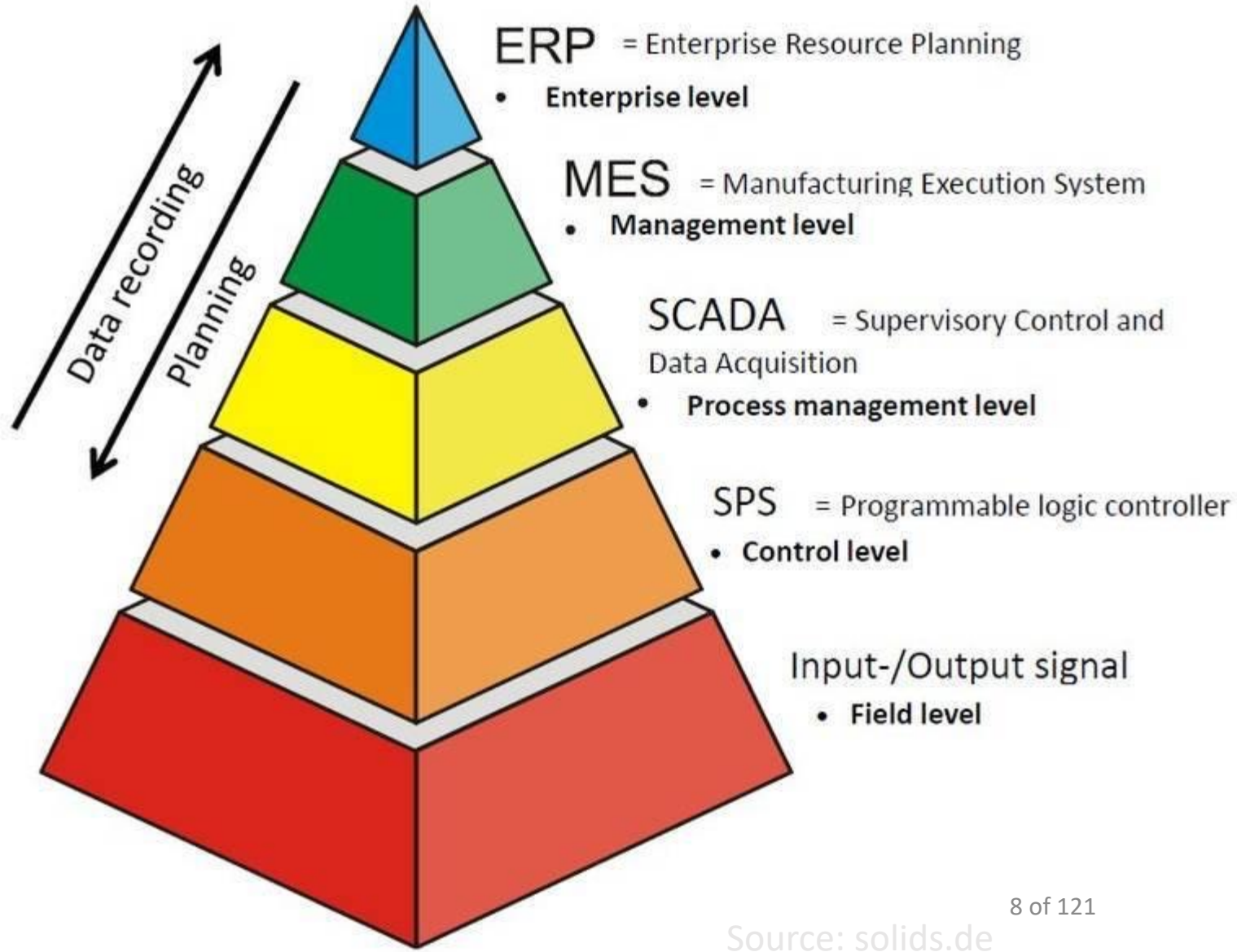
- Industrial Ethernet: a growing market



Industrial Networking Background

- Current organization:

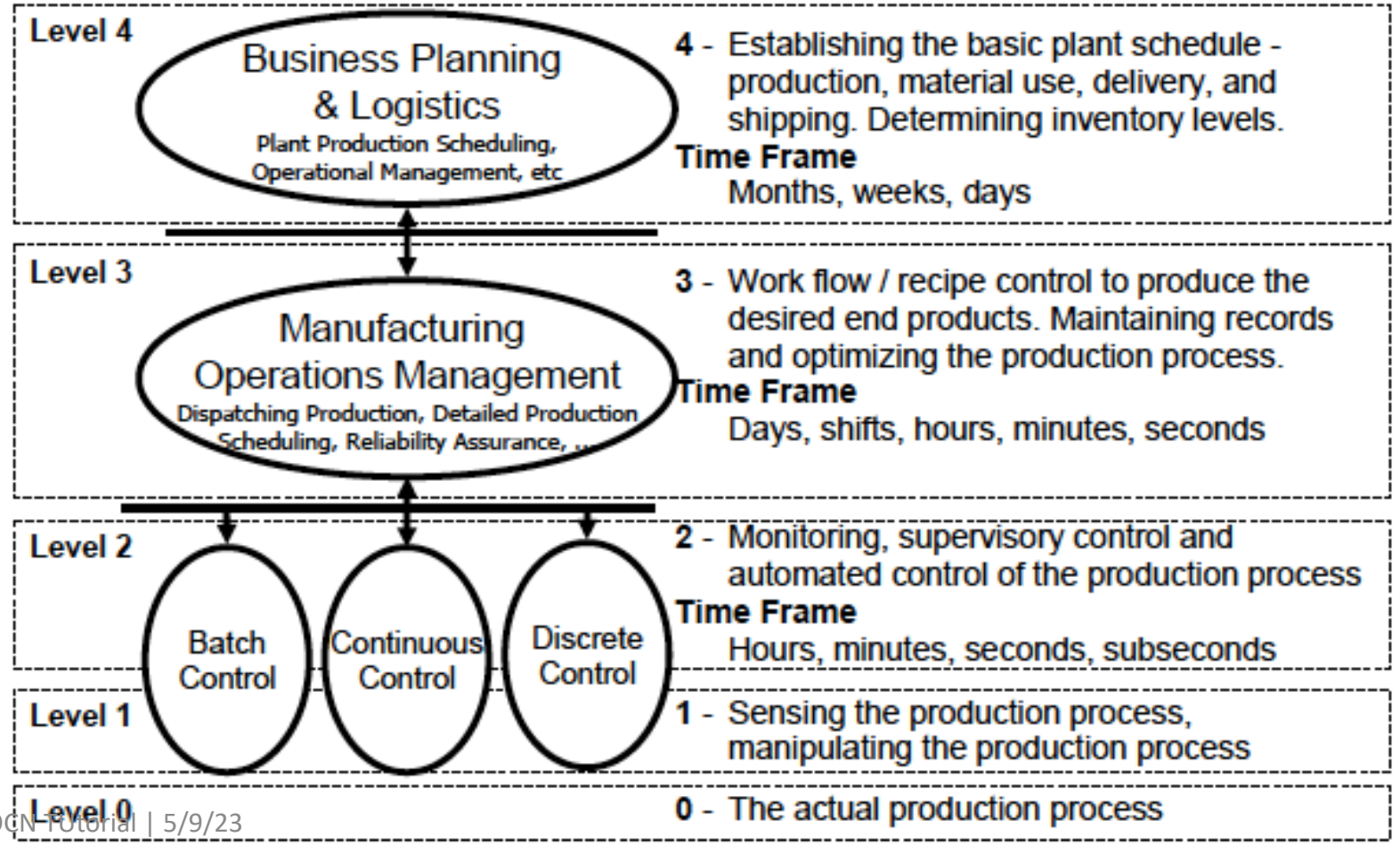
- Goal: to better support this organizational structure



Source: solids.de

Industrial Networking Background

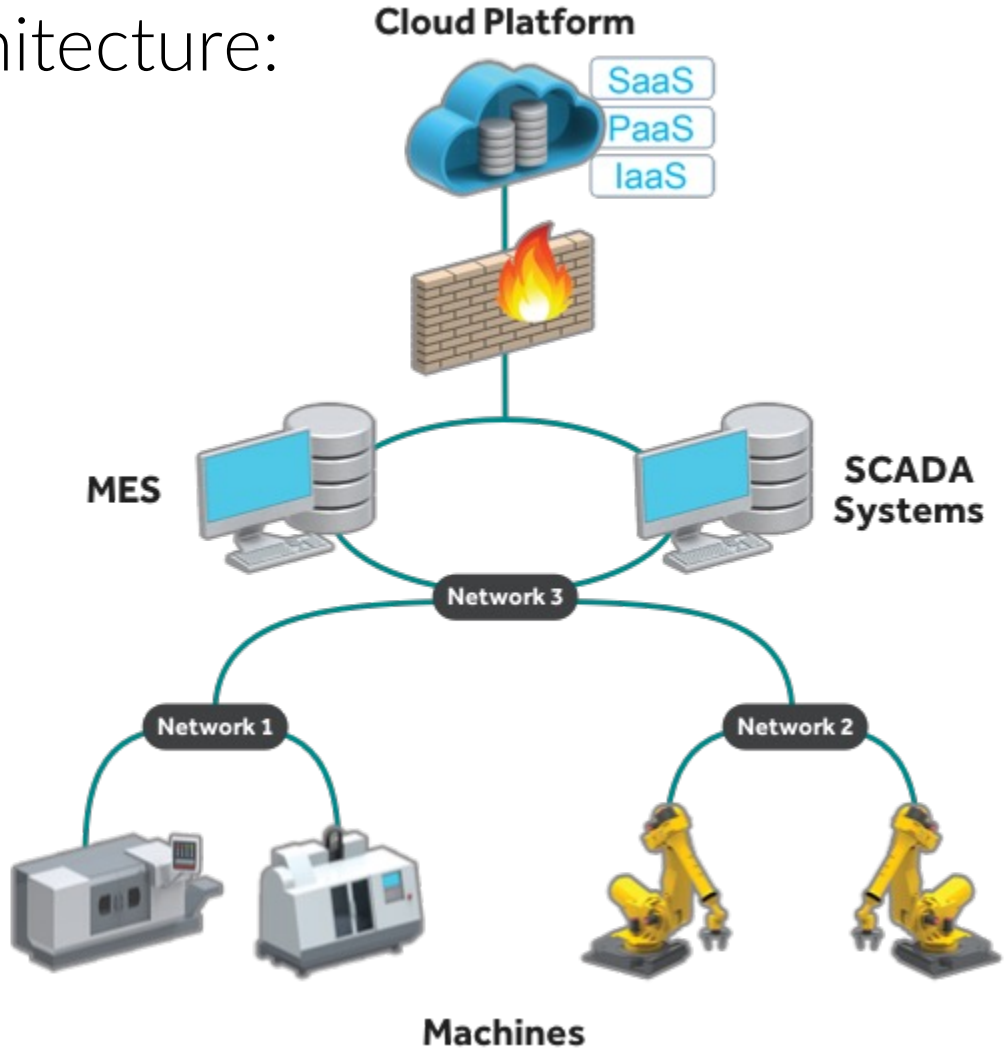
- ISA-95 / IEC 62264: enterprise-control system integration



source: ISA-95/IEC 62264

Industrial Networking Background

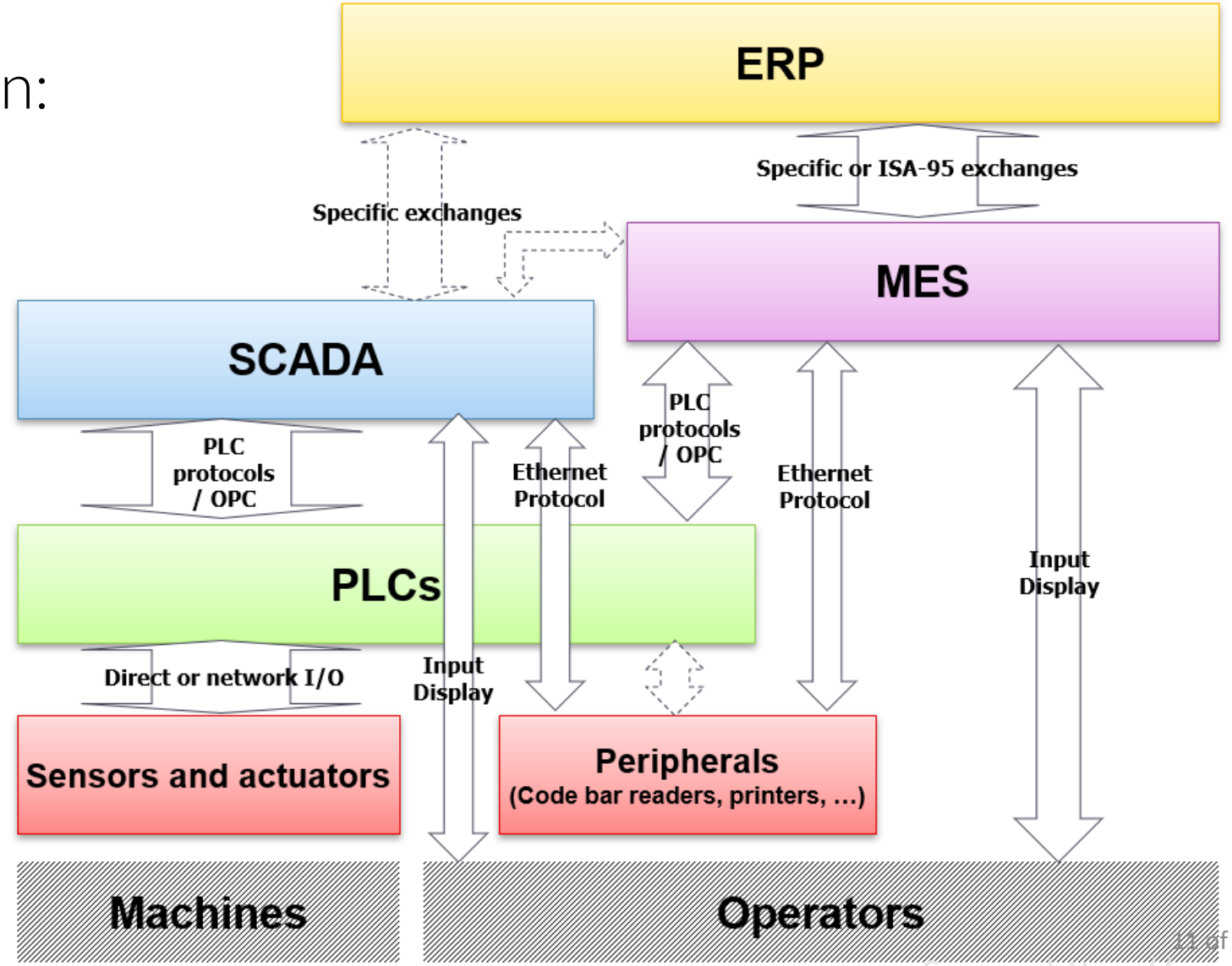
- Typical current architecture:



Source: https://automationforum.co/

Industrial Networking Background

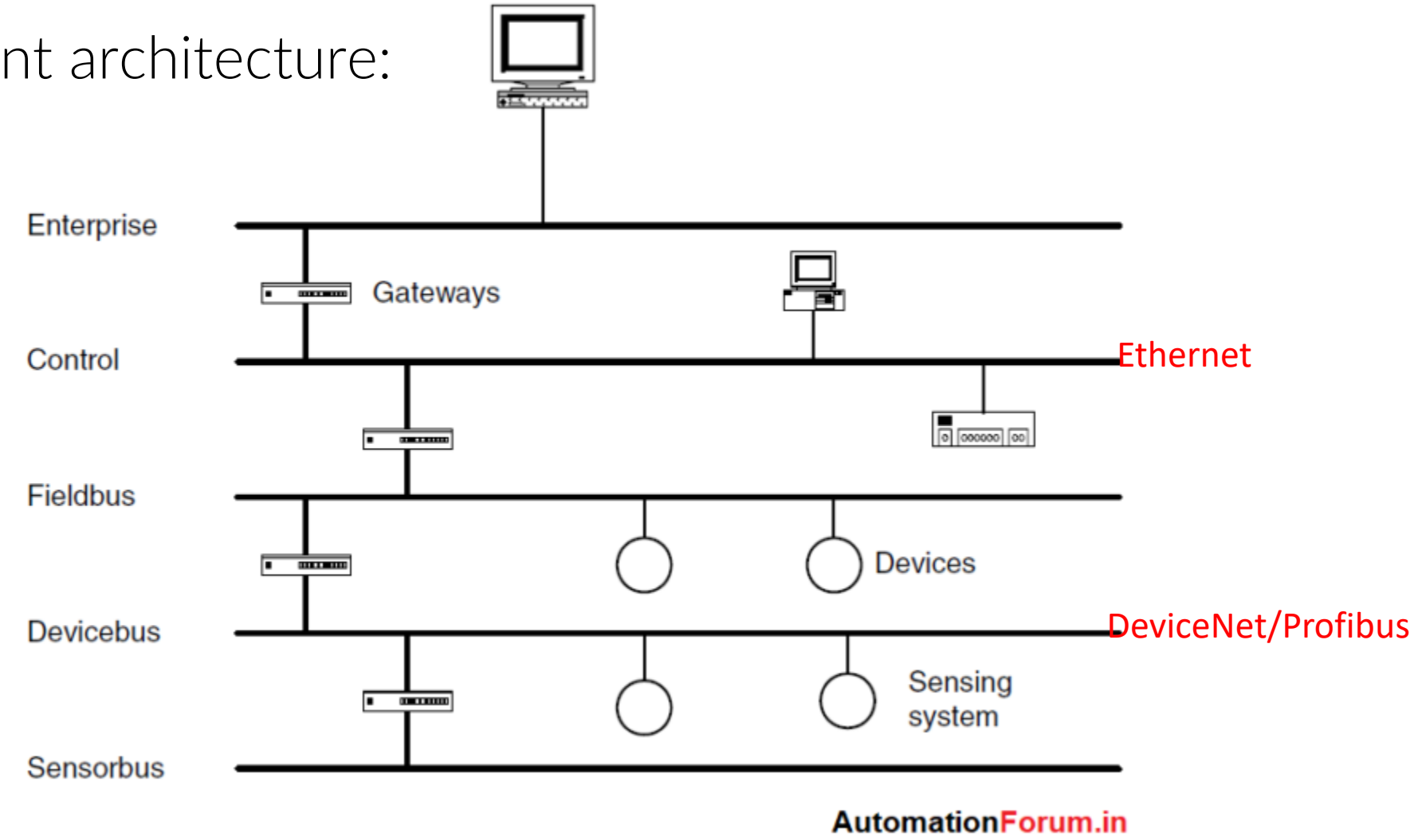
- Functional organization:



Source:https://www.ordinal.fr

Industrial Networking Background

- Typical current architecture:



MES

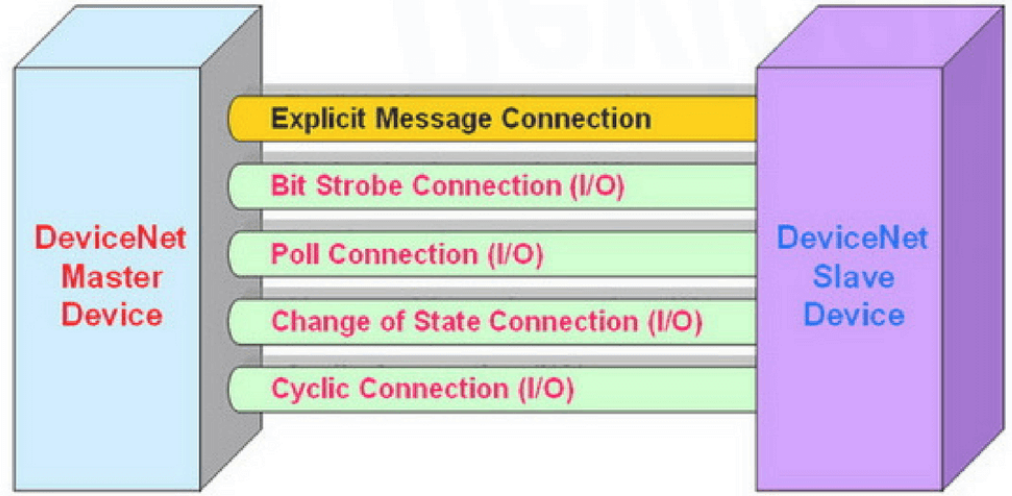
- Manufacturing Execution System
- An advanced system applied to industrial companies with a focus on production
- as a tool aimed at improving Efficiency in Production Plants, by monitoring and collecting data in real time from the machinery and the manual work of the operators
- MES systems: intermediaries between management systems (ERP) and control systems (SCADA, PLCs)

SCADA

- Supervisory Control And Data Acquisition
- A control system architecture comprising computers, networked data communications and graphical user interfaces for *high-level supervision* of machines and processes
- Readings and equipment status reports are communicated to level 2 SCADA
- Data is then compiled and formatted in such a way that a control room operator using the HMI (Human Machine Interface) can make supervisory decisions
- Data may also be stored in database management system
  - to identify trending and other analytical auditing.

Industrial Network Protocols: DeviceNet

- DeviceNet

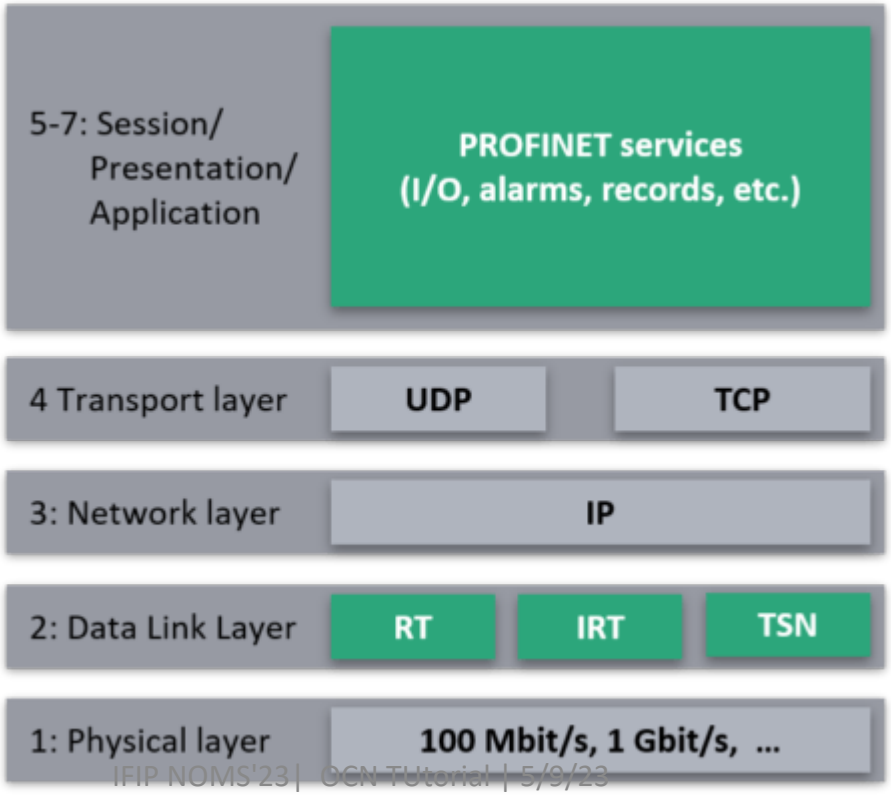


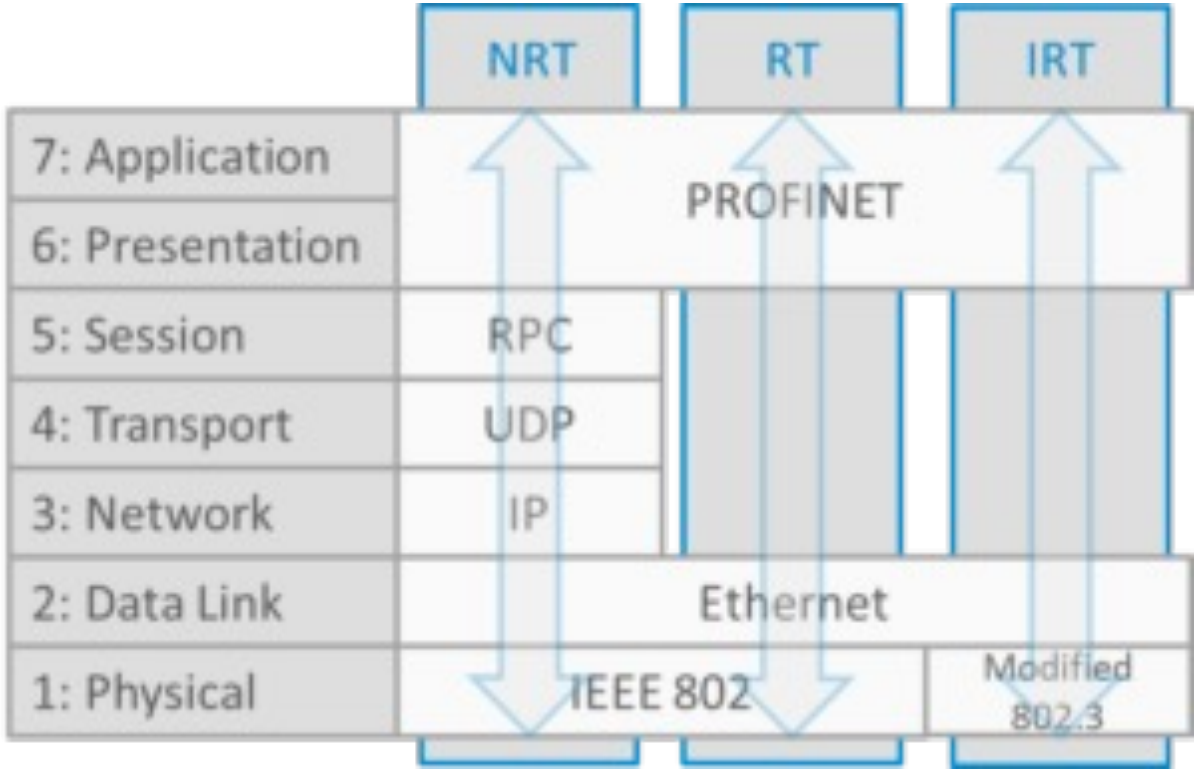
DeviceNet

- A low-level application-layer protocol for industrial applications
  - Created in 1985 by Bosch, then Allen-Bradley-Rockwell Automation created DeviceNet as an application layer protocol based on this standard
- Connection of up to 64 nodes
  - each node can contain simple sensors or process related instrument or programmable logic controller (PLC)
- Trunkline-dropline topology
  - small lines connect devices.and small line connect with main line. These small lines should not run more than six meters from the trunkline
- 500 Kbps – 100 meters/ 250 Kbps – 250 meters/ 125 Kbps – 500 meters

ProfiNet

- Application layer protocol designed to exchange data between controllers and devices in an automation setting
- Introduced in the early 2000s; it is the most well-adopted Industrial Ethernet solution

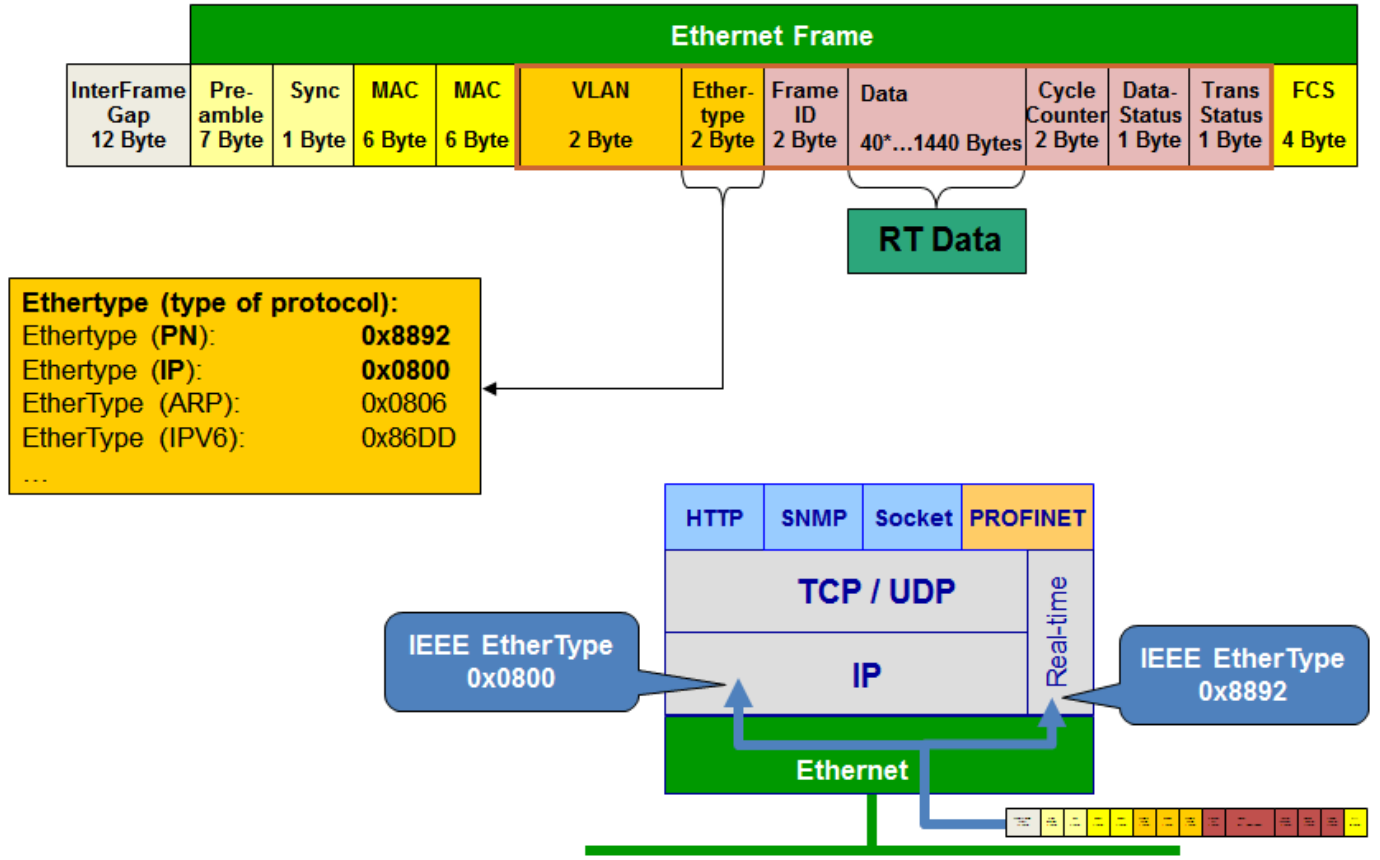




source:https://profinet.com

ProfiNet

- Application layer protocol that offers services
- Three Communication Channels in PROFINET: Real-Time (RT), Non-Real-Time (NRT), and Isoc
- Mostly built on top of IEEE 802 Wireless) for the Data Link Lay
- PROFINET also uses the IP, UD communications
  - Not for RT communications as it
  - RT only within one LAN
  - For NRT communications (diagnc
  - IRT for more strict latency contrc



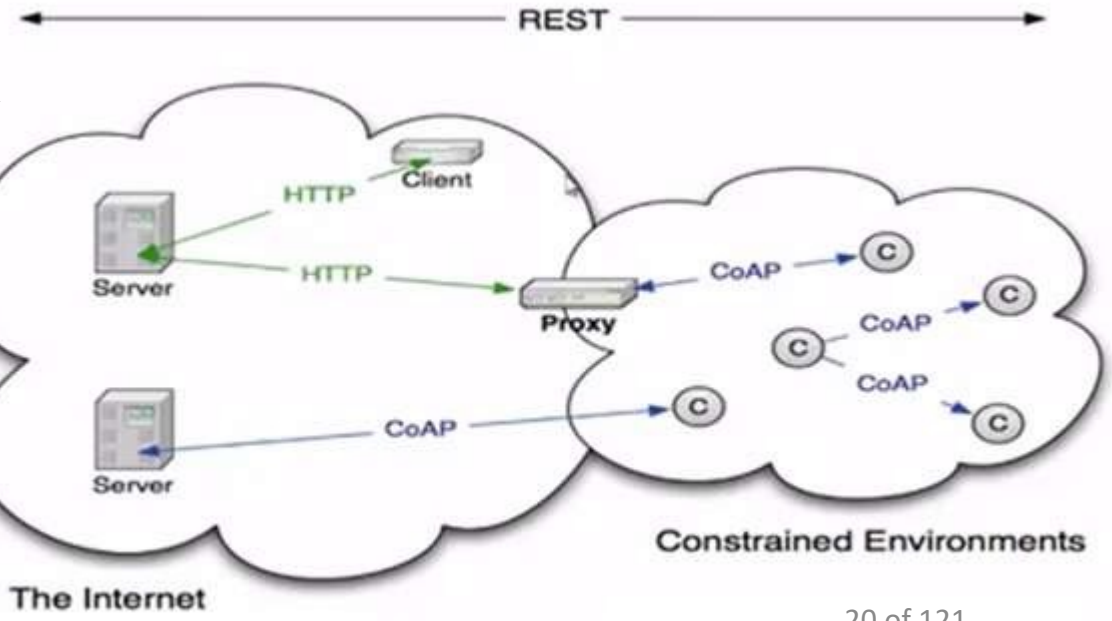
source:https://profinet.com

ModBus

- An open communication protocol developed by Modicon published by Modicon® in 1979 for use with its programmable logic controllers (PLCs)
  - a method used for transmitting information over serial lines between electronic devices
- Defines registers, node ID semantics, client/server relationship, and functions to be transmitted over a serial link
- Limitations:
  - Limited number of data types supported (data types understood by PLCs in the 70s when it was designed); large binary objects are not supported
  - No standard way exists for a node to find the description of a data object (say to learn that a register value represents a temperature between 30 and 175 degrees)
  - Since Modbus is a client/server protocol, there is no way for a field device to get data by the event handler mechanism (except over Ethernet TCP/IP, called open-mbus)
  - Addressing limited to only 247 devices on one data link (again, Ethernet TCP/IP is an exception).
  - Modbus protocol itself provides no security against unauthorized commands or interception of data

COAP

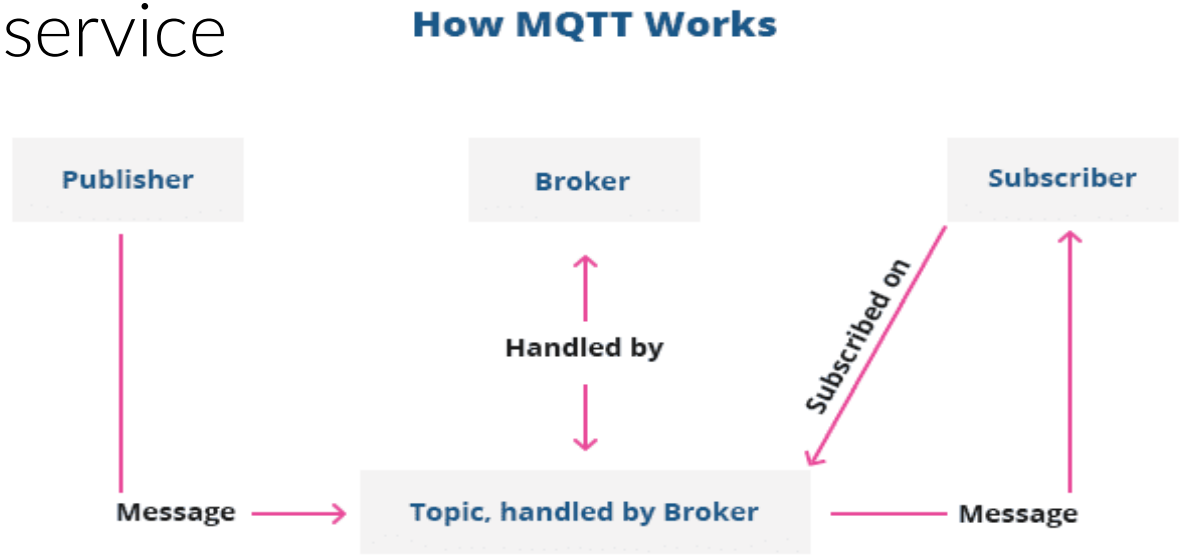
- Constrained Application Protocol (CoAP)
- A specialized Internet application protocol for constrained devices, as defined in RFC 7252
- CoAP is an application-layer protocol that is intended for use in resource-constrained Internet devices
- CoAP makes use of two message types, requests and responses, using a simple, binary header format
  - All instructions should fit within one packet
  - (or L2 frame)
  - UDP based
  - IPv6 or 6LoWPAN



CoAP Architecture

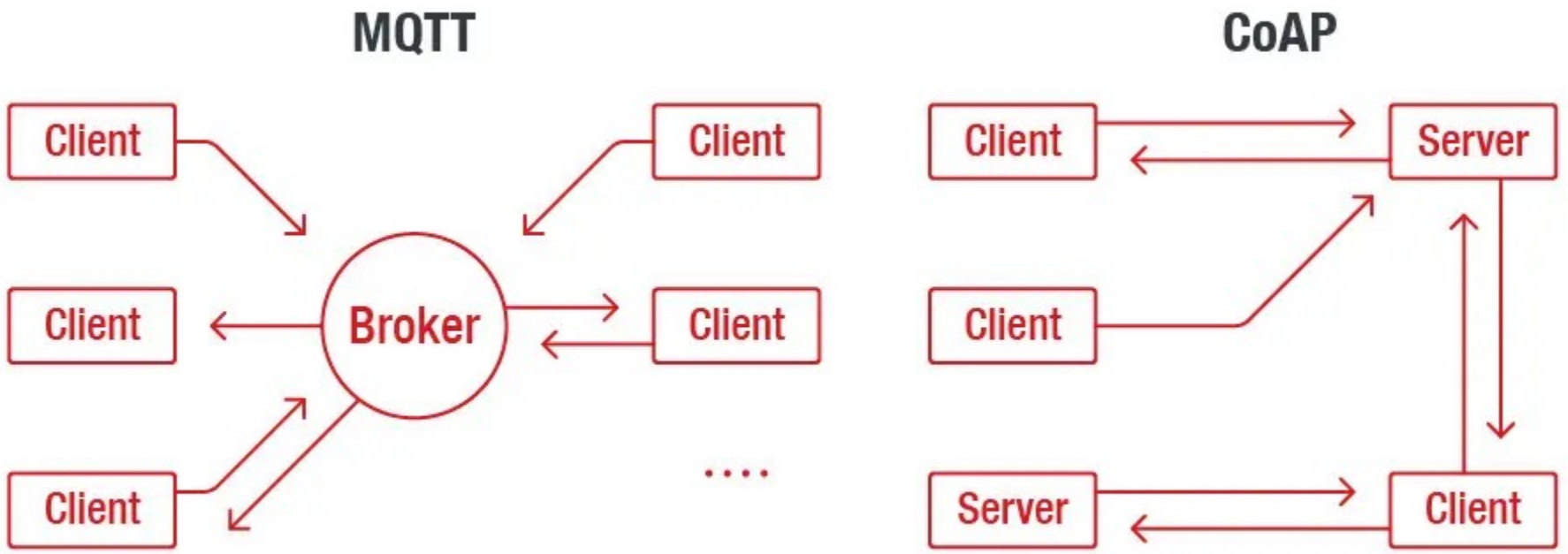
MQTT

- Message Queuing Telemetry Transport
- Invented in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom, now Cirrus Link) as a protocol for minimal battery loss and minimal bandwidth to connect with oil pipelines via satellite
- A lightweight, publish-subscribe, machine-to-machine network protocol for message queue/message queuing service



MQTT vs COAP

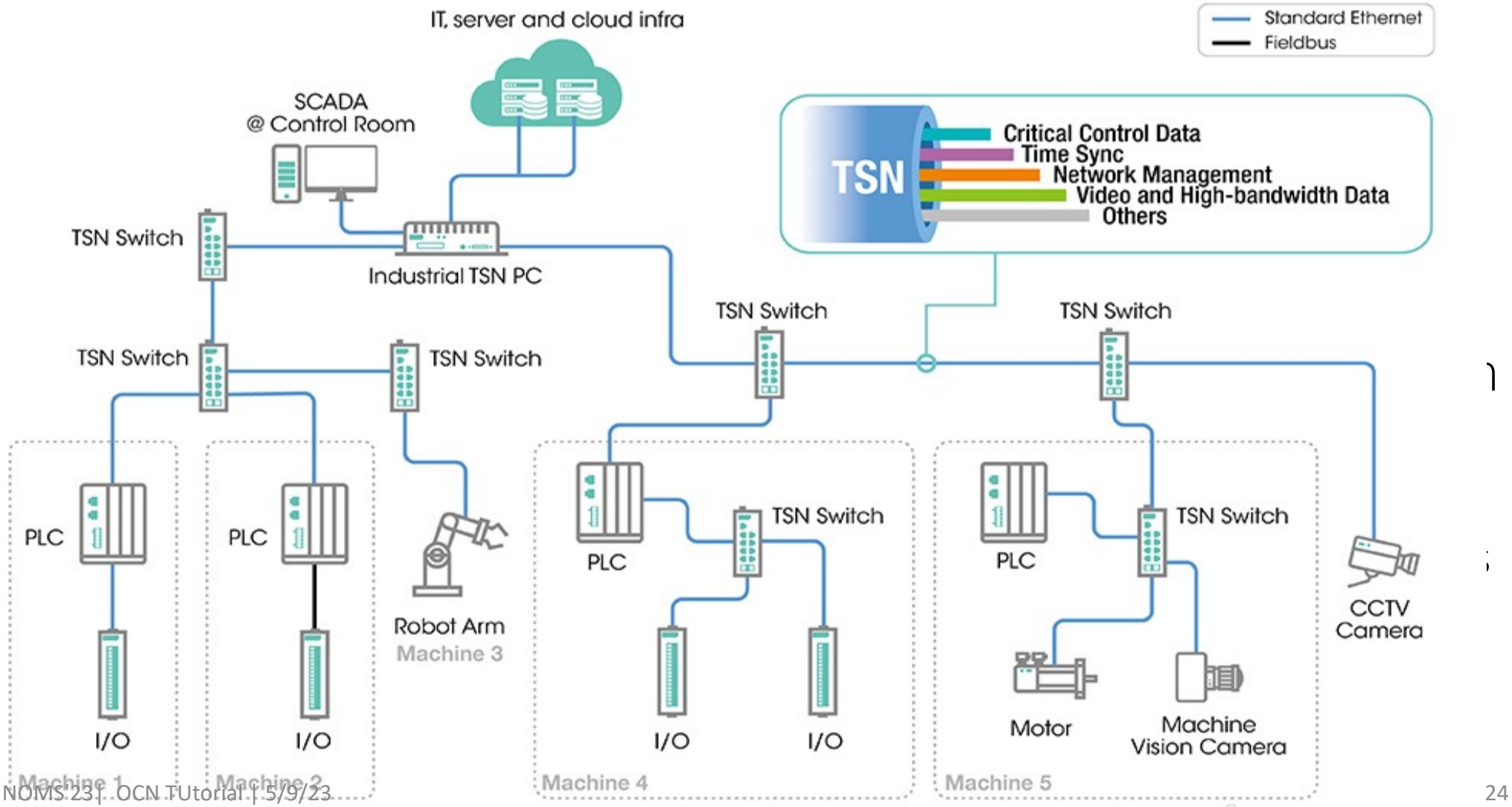
- Interactions



Time Sensitive Networking

- a set of standards under development by the Time-Sensitive Networking task group of the IEEE 802.1 working group
- Ethernet TSN Standards enable Ethernet to become a deterministic networking technology
  - Synchronization of network elements: end-points, switches and gateways
  - Controlled and accountable delay (latency)
  - Prioritization of traffic classes
  - Guaranteed bandwidth reservation
- IRTF DetNet to expand into wider networks

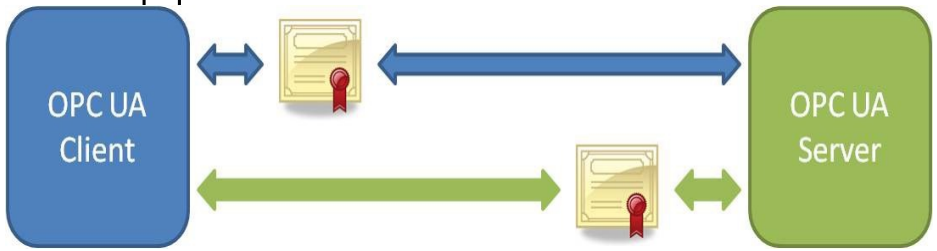
FieldBus/TSN



Source: moxa.com

OPC-UA

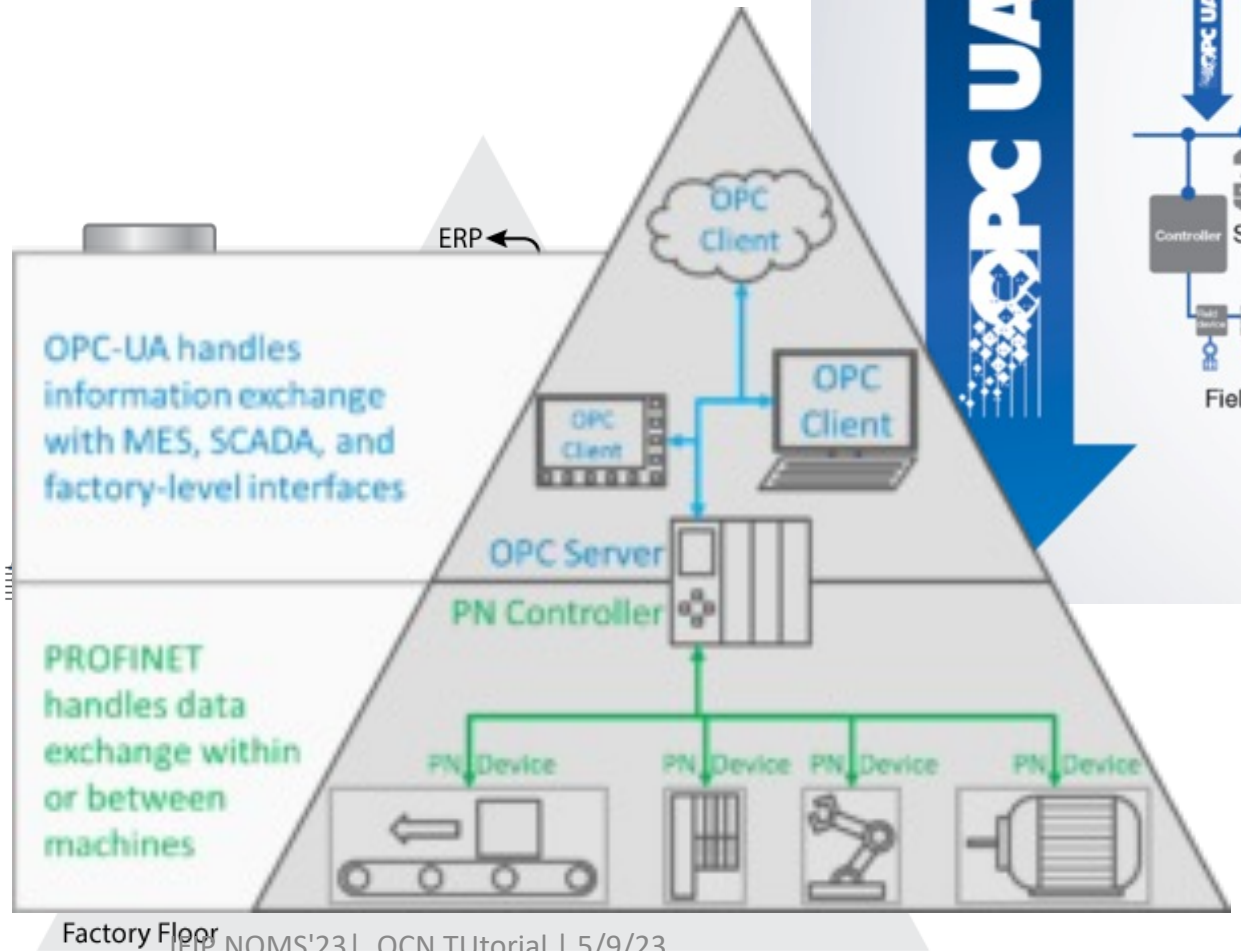
- OPC Unified Architecture (OPC UA) is a machine-to-machine communication protocol used for industrial automation and developed by the OPC Foundation
- OPC originally stands for OLE (object linking and embedding) for Process Control – now Open Platform Communications
- Secure and platform-independent standard for industrial connectivity for devices, automation systems and software applications

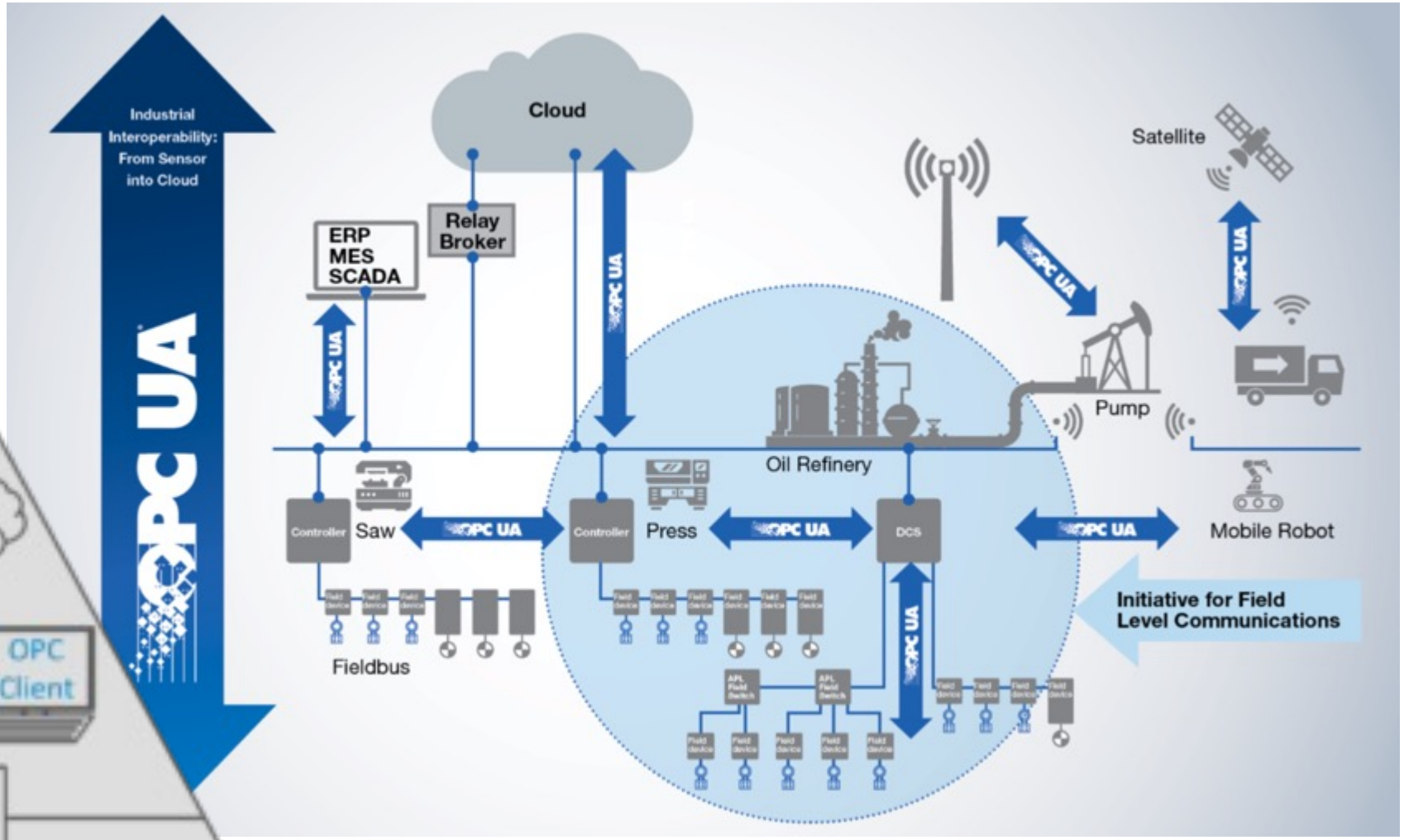


- **OPC is a standard interface to communicate between numerous data sources, including devices on a factory floor, laboratory equipment, test system fixtures, and databases**

Source: ni.com

OPC-UA





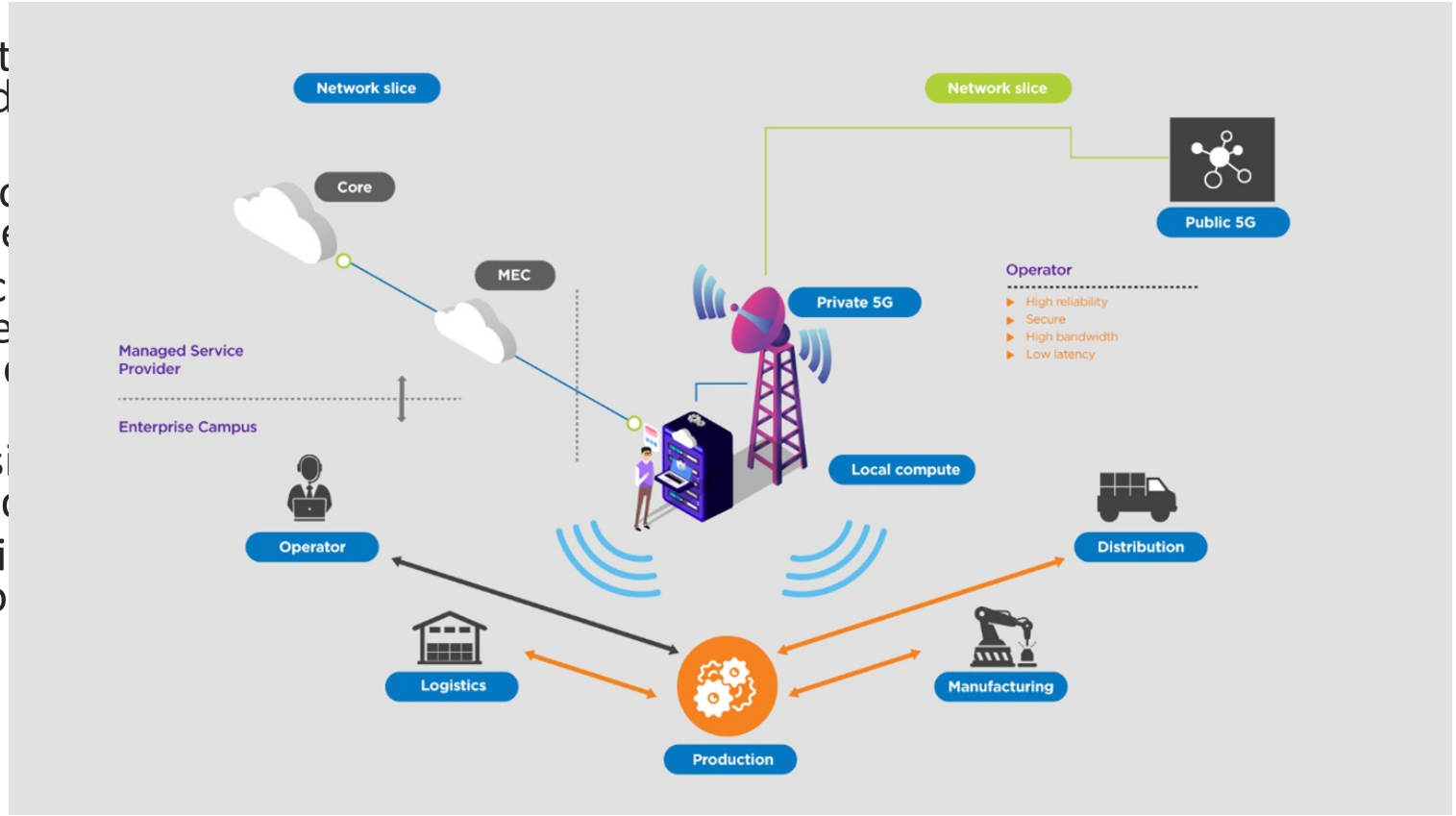
Source: OPC Foundation

Integration of Services

- An illustration: Siemens Industrial Edge platform
- Docker-based container virtualization of automation application
- Container-based approach allows:
  - Applications isolation and scaling
  - Support for self-developed or 3rd party Apps to the shopfloor
  - Leverage added value with production data by integrating it directly into automation systems
- BUT: wall-garden under control of one company
- BUT: edge platform still on the local LAN

Industrial 5G

- Private 5G networks allow deployment of cellular in the context of OCN
  - Open source deployments: ONF Aether, Facebook Magma
- 5G provides low latency, high bandwidth and wider scale
  - Production line efficiency: 5G is more scalable for larger facilities and has higher bandwidth density than wifi
  - Autonomous guided vehicles (AGV): Privat and reliable mobile handovers across prod warehousing.
  - Machine to machine (M2M) communicatic private 5G network, network slicing enable
  - End-to-end logistics: support low cost trac environment of finished goods, parts, asse national and international logistics with a and public 5G networks
  - Predictive maintenance: rapid data acquisi rates. This enables systems to provide pre
  - Remote heavy equipment control: Heavy i cranes, or earthmovers are mostly outdoo private 5G to function well.

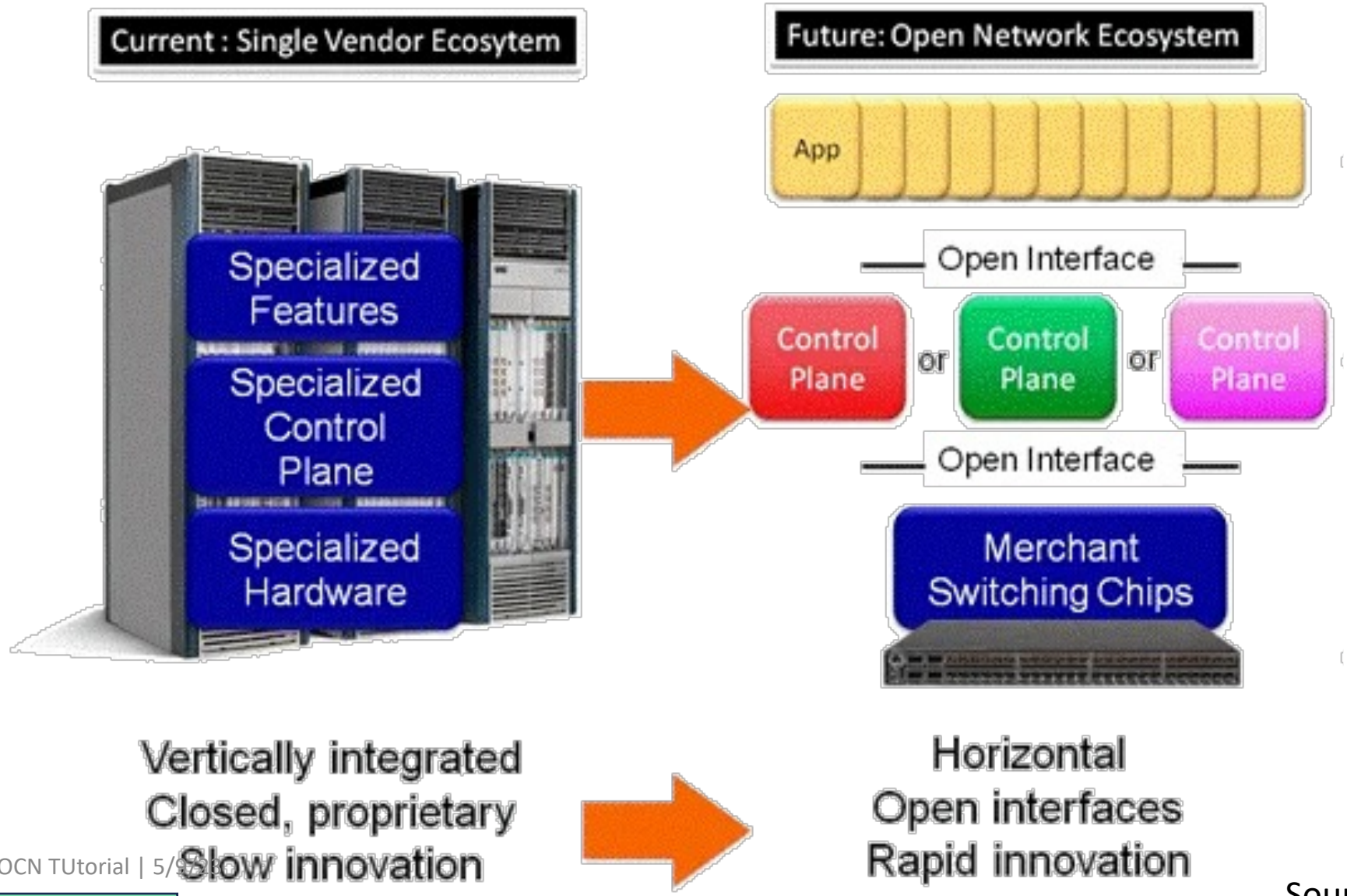


Source:hcltech.com

Summarization of shortcomings/issues

- Short range for DeviceNet + low number of nodes
- ProfiNet RT/IRT within the same LAN: small scale
- Plethora of protocols
  - With some incompatible abstractions
- Vertical integration and siloes: walled gardens
  - Control by a company through validation of services
- Integration of new technologies such as 5G
  - Global footprint for industrial networks

The comparison with SDN



Source:upc.edu

Horizontal Decomposition of OCN

- Similar trends in OCN
- Vertical integration of application
- Limited flexibility in adding new
  - Devices (need to match the other proprietary protocols)
  - Services (need to be supported by the hardware)
  - Protocols (proprietary and closed)
- The same opportunity exists:
  - Open to modularity
  - Agility in deploying new devices, functions, etc.
- With some great challenges
  - Security

FUTUREWEI
Technologies

Use Case – Factory Automation

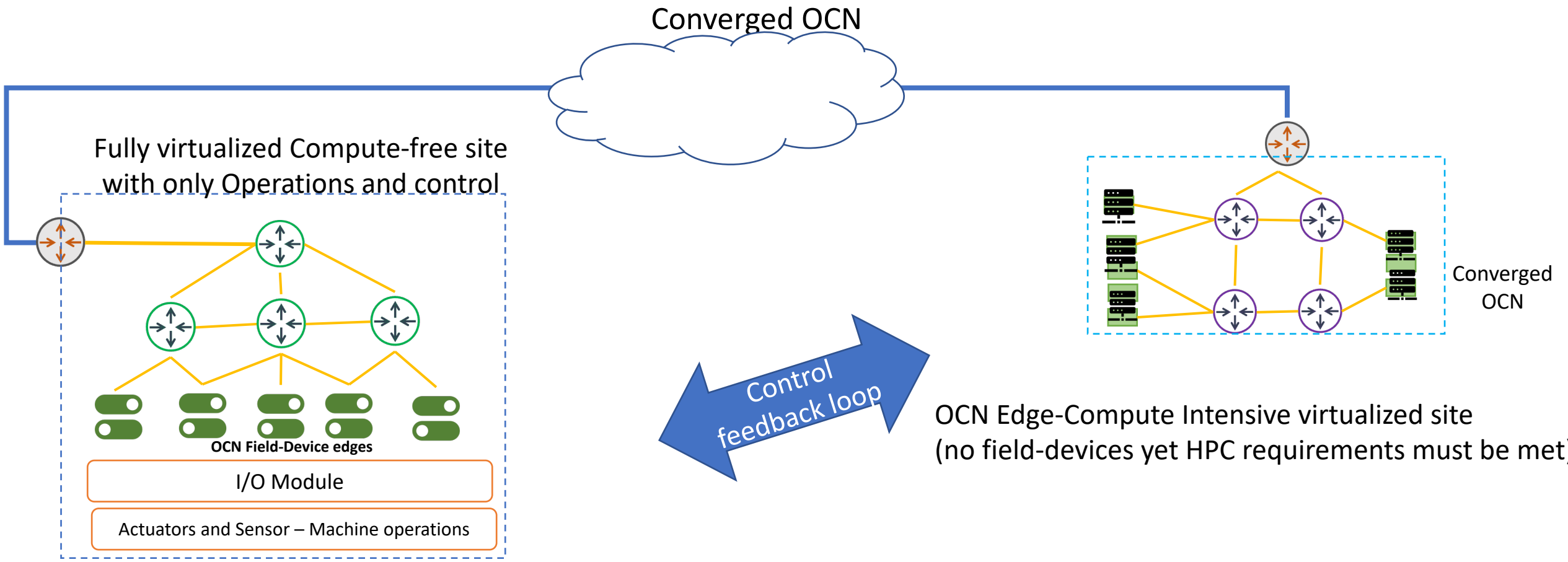
Use Case:

- Use cases to support:
  - More open architecture, using common standardized solutions
  - More scalable architecture, to scale up from local domain to multiple domains or wide area domains
  - More flexibility in organizing industrial networks
  - The convergence of IT and OT -> to speed up adoption of IT practices for network and software development
  - The convergence of protocols to unify the network; inter-operate; increase markets; etc.
  - The deployment of open interfaces that enable innovation

Use Cases:

- Protocol Convergence
- Self-Configuration of Industrial Networks
- 5G Private Networks
- IIoT
- Large Volume Application
- Remote Control
- Determinism
- Industry 5.0
- Virtual PLC
- Simplification of OT/IT Integration
- Smart Contract

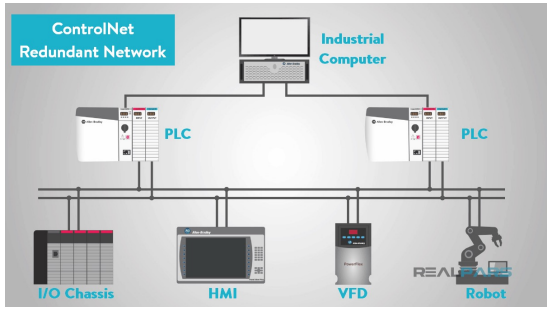
IT/OT Converged OCN for remote operations

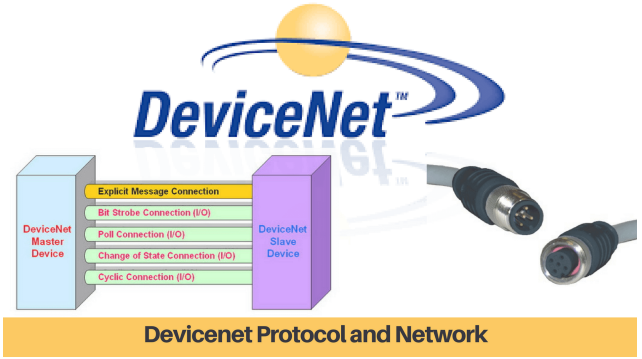


FUTUREWEI
Technologies

Industrial Legacy Protocol Convergence

Protocol Convergence

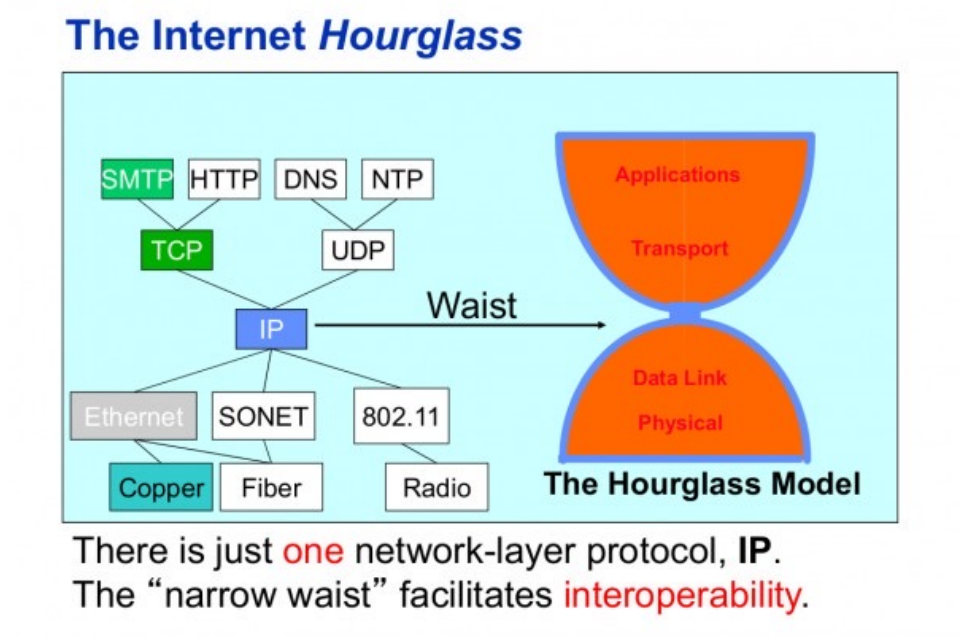




- These protocols were developed decades ago and are still widely used in industrial control systems for tasks such as data acquisition, control, and monitoring.
- Integrating legacy protocols into IIoT systems can be challenging because these protocols often use proprietary communication methods and lack the necessary security features to protect against cyber attacks.

Success of Narrow Waist

- It allows for interoperability between different devices and networks
- It allows for flexibility and scalability in the design of the protocol stack.
- By keeping the lower layers of the stack simple and generic, it becomes easier to add new functionality and features to the higher layers without having to modify the lower layers.



We need a glue to connect these different legacy industrial networks together, similar to the way that the Internet became successful once a common narrow waist was defined and provided.

Evolution of Process Control Network Convergence

Often deterministic connectivity via proprietary, isolated networks

Unified Infrastructure, w/ Loose determinism

Bring back Determinism

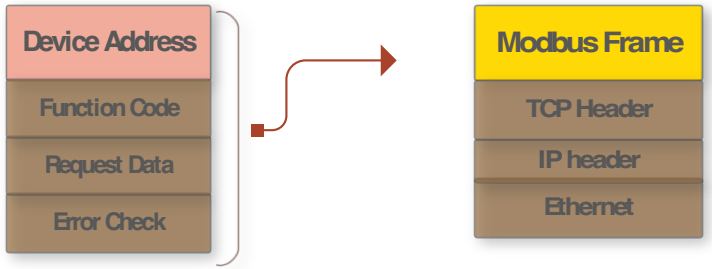
Enable Digital Tranformation and Automation in Operations

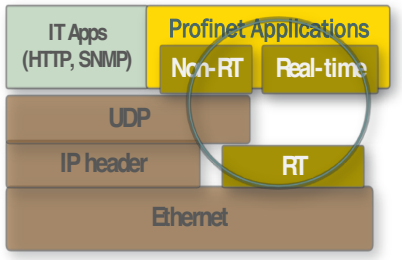
Traditional OT Networks

Converged Network Technologies

Process Control and Automation

Enabling control in Data Ecosystem





Convergence of OT/IT

- Beyond collection and analyses, bring changes to device/environment.

Time-Engineered algorithms by design...

- Support Cloud or Edge networks for compute capabilites.

Modbus Profinet,… and serial I/O

TCP/IP/Ethernet
no determinism

TSN , DetNet
Smaller-scale/Limited use technologies

Fully formed Automation using OCN

Requirement

- Interoperability
  - Interoperability is a key requirement for protocol convergence
  - Using common protocols or framework for different types of devices and systems to communicate with each other.
  - Importance of standardized protocols for interoperability

- Scalability
  - Protocol convergence should be able to scale to accommodate growing numbers of devices and systems
  - The capability to process significant volumes of data and manage multiple devices in real-time.
  - Requirement for adaptable and scalable frameworks that can be expanded as needed.

- Ease of Use
  - Need for protocols that are simple to configure and user-friendly.
  - Ability to integrate with existing systems and tools

- Security
  - Secure protocols that can safeguard against cybersecurity threats.
  - The capability to authenticate the identity of devices and systems and encrypt information.

FUTUREWEI
Technologies

Cloud-based PLCs

Programming Logic Controllers (PLCs)

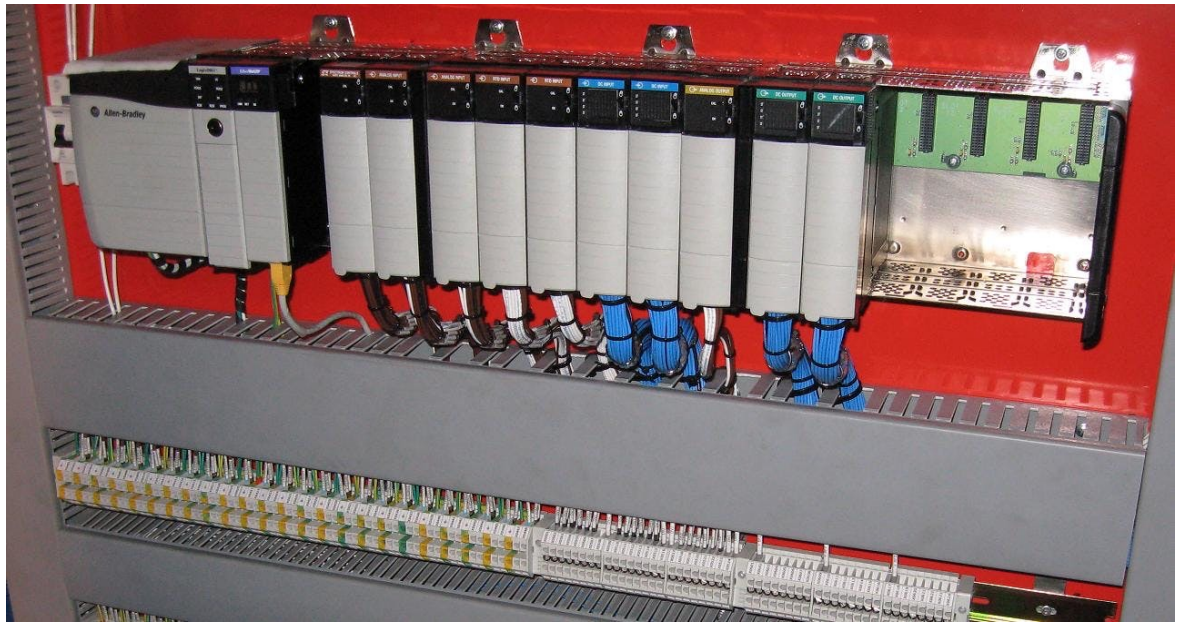
- Programming Logic Controllers (PLCs) are digital computers used in industrial settings to automate processes and control machinery.

- They are designed to withstand harsh environmental conditions, such as extreme temperatures, humidity, and vibration, making them ideal for use in factories and other manufacturing facilities.

Ref: https://roboticsandautomationnews.com/2020/07/15/top-20-programmable-logic-controller-manufacturers/33153/

Basic Components of a PLC

- PLCs consist of a central processing unit (CPU), memory, input/output (I/O) modules, and communication interfaces.
- The CPU is responsible for executing the program logic, which is stored in memory and consists of a set of instructions that tell the PLC how to respond to input signals and control output signals.
- The I/O modules are used to interface with sensors and actuators, while the communication interfaces allow the PLC to communicate with other devices on the network.
- PLCs are programmed using specialized software that allows engineers and technicians to create and edit the program logic. The software typically includes a graphical user interface (GUI) that allows users to create a visual representation of the control system and program the logic using ladder logic diagrams, function block diagrams, or other programming languages.



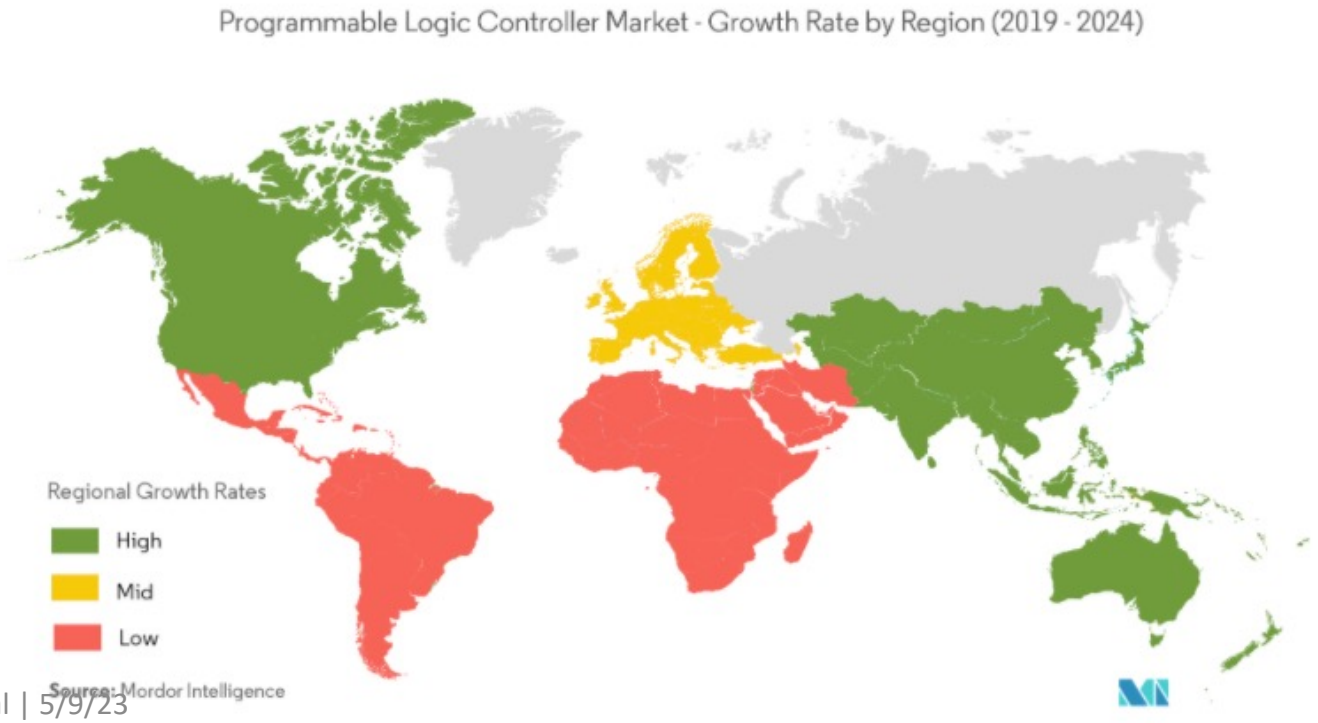
Allen-Bradley PLC rack

ref: https://www.machinedesign.com/learning-resources/engineering-essentials/article/21834250/engineering-essentials-what-is-a-programmable-logic-controller

Growth of PLCs

- According to research by Mordor Intelligence, the PLC market was worth almost $4 billion ($3,897.36 million, says Mordor) in 2019. It is expected to reach $4.3 billion ($4,292.66 million) by 2025, at an annual growth rate of 3.7 percent over Mordor's forecast period of 2020-2025.

- A large portion of the growth is expected to come from the automotive manufacturing sector, where PLCs first emerged, and the sector that uses more industrial robots than any other.



Advantages of PLCs

- PLCs can respond to input signals much faster than traditional control systems. This allows them to make real-time adjustments to the control system, which can improve efficiency and reduce downtime.
- PLCs can be easily reprogrammed to adapt to changes in the control system or to implement new features. This means that they can be used in a wide range of applications.
- PLCs can help to reduce operating costs by improving efficiency and reducing downtime. They can also reduce the need for manual intervention, which can help to reduce labor costs.
- PLCs can be easily integrated into larger control systems, and they can be used to control a wide range of machines and processes.

Disadvantages of Legacy PLCs

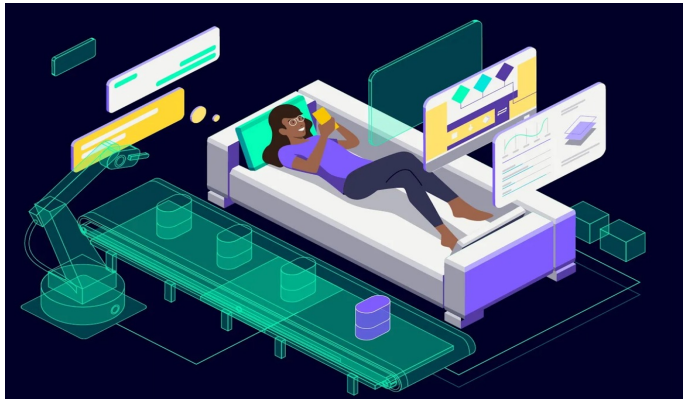
- Cost: The initial cost of purchasing and installing a PLC system can be relatively high, which can be a barrier for organizations with limited budgets.
- Complexity: PLC programming requires specialized knowledge and skills that may not be readily available.
- Limited Flexibility: Once a PLC system is programmed and installed, it can be difficult to modify or adapt to changing process requirements.
- Compatibility Issues: Different PLC brands and models may have different communication protocols and programming software, which can create compatibility issues when integrating with other control systems or devices.

Hazardous and rural environments where human access is limited or not feasible.

- Some examples of hazardous environments in industrial IoT include:
  1. Chemical Plants: Chemical plants often contain hazardous substances that can pose a risk to human health and safety.
  2. Oil and Gas Industry: The oil and gas industry operates in offshore oil rigs, where workers are exposed to extreme temperatures, high pressures.
  3. Mining Industry: The mining industry operates in underground mines, where workers are exposed to dust, gases.
  4. Nuclear Power Plants: Nuclear power plants operate in hazardous environments, where exposure to radiation can be dangerous for workers.
  5. Food and Beverage Industry: The food and beverage industry operates in environments where contamination can pose a risk to human health.

- In order to avoid the human involvement in the interaction with PLCs on the field, Cloud-based PLCs are a relatively new trend in industrial automation that leverage the benefits of cloud computing and the internet of things (IoT) to provide remote access and control of PLC systems.

Ref: https://www.ie-bearing.com/news/remote-condition-monitoring-motors-using-power-internet/



Ref: https://www.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500/virtual-plc.html

Examples of Cloud-based PLCs

- Hazardous Environments: Cloud-based PLCs can be used to control and monitor industrial processes in hazardous environments, such as chemical plants, oil rigs, and mining sites. Human exposure to hazardous materials and conditions can be minimized, reducing the risk of injury or illness.

- Remote Monitoring: Cloud-based PLCs can be used to remotely monitor and control processes in rural environments, such as agricultural fields or remote water treatment facilities.

- Unmanned Vehicles: Cloud-based PLCs can be used to control unmanned vehicles, such as drones or autonomous vehicles. Human operators can be kept out of harm's way while still being able to collect valuable data and perform important tasks.

- Emergency Shutdowns: Cloud-based PLCs can be used to automatically shutdown industrial processes in the event of an emergency, such as a fire or explosion. By using Cloud-based PLCs to automate these shutdowns, human response time can be minimized, reducing the risk of injury or damage to equipment.

Ref: What is IIoT (Industrial Internet of Things)? https://www.youtube.com/watch?v=PovkWOJGSnM&t=141s

Requirements

- Reliability : Industrial networks must operate continuously, reliably, and without interruption to ensure that critical processes and operations run smoothly.
  - Protocols that guarantee high availability and resilience to faults are necessary.
  - Can detect and recover from faults in the network infrastructure, ensuring that the network continues to function correctly even if individual components fail.
  - Effectively recover from failures and manage network congestion. Network congestion can lead to delays, dropped packets, and other issues that can impact the performance and reliability of the network.

- Low latency : Real-time control and monitoring of industrial processes require minimal latency, which is the time it takes for data to travel from one point in the network to another. Low latency ensures that control signals and data are transmitted quickly and that the network can respond rapidly to changes in the environment.
  - Optimize network performance by minimizing overhead and reducing the amount of data that needs to be transmitted.

- Guaranteed levels of service for different types of traffic
  - Different types of traffic have varying requirements for latency, reliability, and throughput.
  - The capability to prioritize traffic and guarantee prompt transmission of critical information is essential to ensure that industrial processes and operations run smoothly.

FUTUREWEI
Technologies

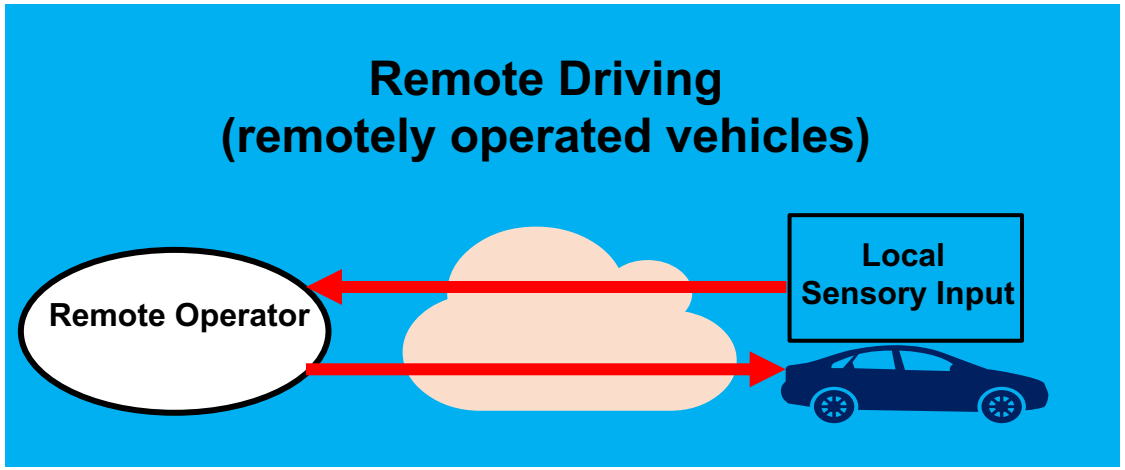
Remote Driving

Autonomous driving and remote driving

- Autonomous vehicles (AV) have made great progress in the recent years, which rely on numerous well-placed sensors.
- Autonomous vehicle can be controlled by its own central computer.
- SAE International's new standard "J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems" defines six LoAs (Level of Automation) [SAEJ3016], including full automation (level 5), high automation (level 4), conditional automation (level 3), partial automation (level 2), driver assistance (level 1), and no automation (level 0)
- Although each vehicle manufacturer has been taking its best effort of making progress in increasing the level of automation, the current automated vehicles by themselves can only fit into the SAE classification 2 or 3.
- AVs may fail short in unexpected situations. In such cases, it is desirable that humans can operate the vehicle manually to recover from a failure situation through remote driving.
- Until the autonomous technology becomes mature enough to be level 5, the experts suggest AVs should be backed up by tele-operations.

Remote driving further assists the autonomous driving

- Remote driving is a mechanism in which a human driver operates a vehicle from a distance through communication networks. Remote driving leverages the human driver's advanced perceptual and cognitive skills to further assist the autonomous driving:
  - Perception failure at night or under challenging weather conditions
  - Confusing or malfunctioning traffic lights,
  - Unrecognizable traffic signs due to corrosion or graffiti
  - Confusing detour signs or complex instructions temporarily ordered by police officers
  - Complex or confusing parking signs, which might be handwritten and hard to be understood by computers.
- With remote driving being added to the AV control loop, passengers could feel safe enough.



Remotely operated vehicles for personal transportation services

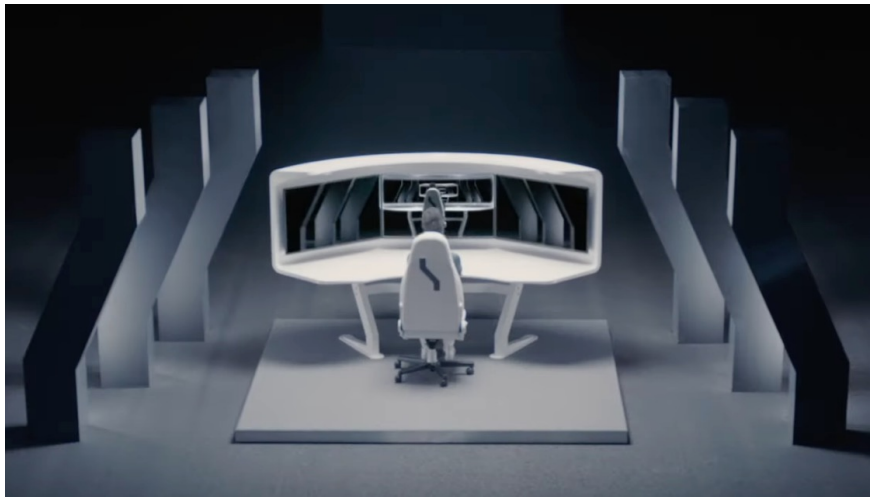
- Vay, a Berlin-based startup has launched a fleet of taxis controlled by remote teledrivers .
- The concept behind Vay is that when you order a Vay, one of teledrivers is tasked to navigate one Vay to your pickup location. Then you take control the Vay.
- After you reach your destination, the teledriver takes control of the Vay and deliver it to the next nearby customer.
- During the whole transaction, the remote driving takes place for Vay delivery.

https://vay.io/

Remotely operated trucks

- Remotely operated trucks could possibly eliminate the threats to road safety, driver/passenger safety that are caused by fleet driver fatigue during long drives.

- Remotely operated vehicles are also particularly useful compared to autonomous truck in hazardous environment.



https://www.einride.tech/

Collison avoidance in remote driving

A remotely controlled vehicle needs to transit necessary data in high volumes to the remote operation center which might be located in edge cloud or central cloud.

- Image capture, encoding, decoding and display: 100 ms
- Remote driver's reaction time: 100 ms;
- Total transmission time in the network: 50-200 ms, which includes the time for the image data to reach the remote driver as well as the time for the command to reach the vehicle .
- Total: 250-400 ms.

Speed	Collision Avoidance Distance
5 km/hour = 1.4 m/sec	1.4*0.4 = 0.56 m
30 km/hour = 8.4 m/sec	8.4*0.4 = 3.36 m
60 km/hour = 16.8 m/sec	16.8*0.4 = 6.72 m
80 km/hour = 22.3 m/sec	22.3*0.4 = 8.92 m
120 km/hour = 33.4 m/sec	33.4*0.4 = 13.36 m

Network requirements

- The networking services shall support multiple concurrent flow streams at high data rates and volumetric data transmission from vehicles with high mobility.
- The networks shall deliver services with service level objectives, specifically latency objectives. The latency objectives are required to be precisely guaranteed and highly reliable, not just "optimized" but quantifiable.
- The network shall be able to identify the packets which carry urgent information and treat them in a differentiated manner with highest priority.
- The networking services shall reduce and even avoid dropping/re-transmission of packets with high significance. Packet loss of certain urgent packets are not permissible in the network.

FUTUREWEI
Technologies

OCN Model

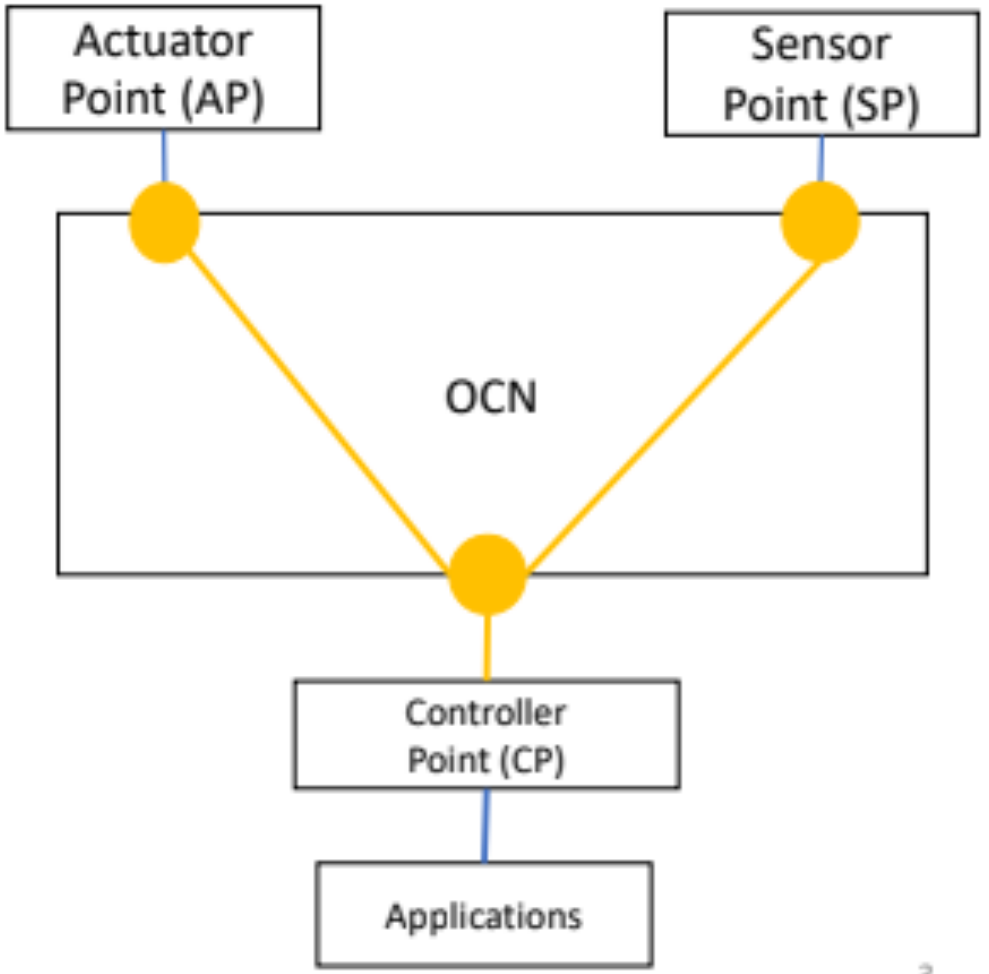
What is OCN?

- An attempt to generalize a network for control systems based on two aspects
  - That the field devices do not operate on their own.
    - Instructed by controllers to do certain actions and read certain sensory data.
    - Those controllers can be of any form-factor (small, large, lots of processing power, etc.)
  - For a given application the behavior and operations are well-defined.
    - Characteristics of sensors are quite specific. Emitting data towards controller.
    - Actuator properties – Write and readback type of commands.

- Motivation:
  - There are growing number of “remote operation” type of use cases in which there’s network between field devices and the controller.

OCN Reference model

- We can start by thinking of OCN as a abstraction of control systems.
- With key logical role-based Referenc points
  - Actuators
  - Sensors
  - Controller
- Then standardize
  - (a) interfaces and
  - (b) common message types and
  - (c) corresponding network-constraints

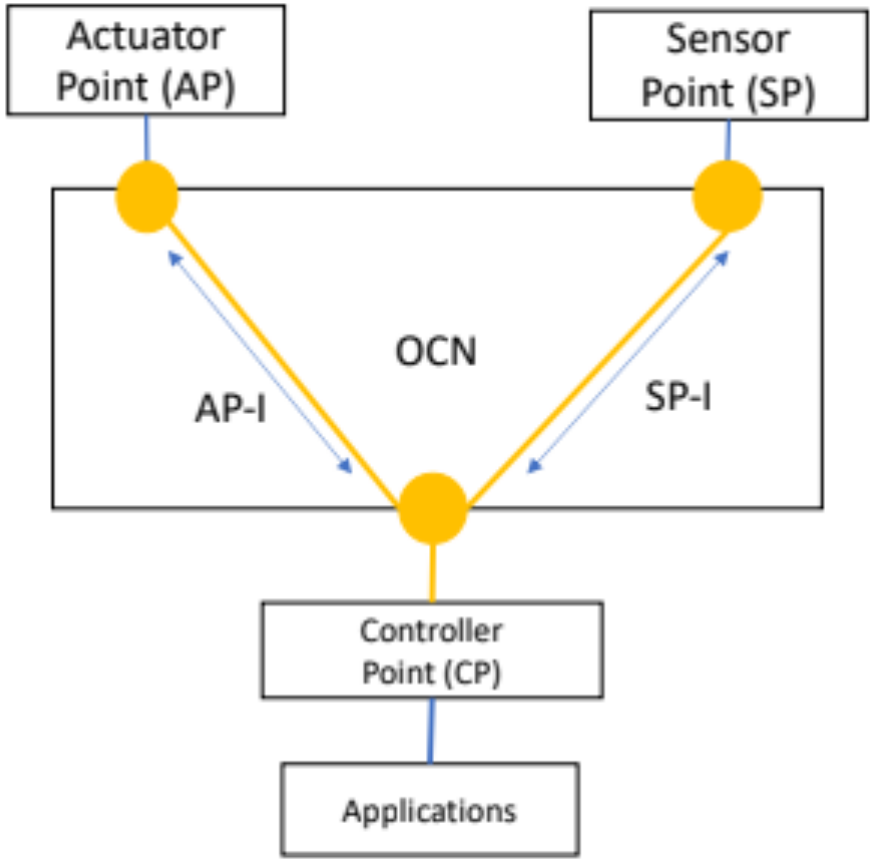


OCN Reference model: Interfaces

Nomenclature used for Reference points

- Actuator Point (AP)
- Sensors Point (SP)

- Sensor Point Interface (SP-I) CP<-> SP
  - Expresses sensor type behavior
  - Concern with data emitted by sensor
  - Message severity, periodicity, etc.
  - Solicit reads from sensors
- Actuator Point Interface (AP-I) CP<->AP
  - Typical actuating point.
  - Data the brings out changes to physical environment or mechanical movements
  - Precise time-based message delivery, feedback control loop, open control loop

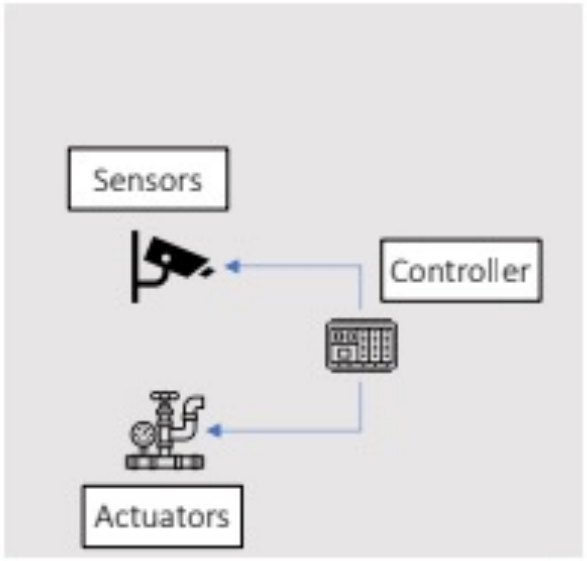


OCN Messages and network characteristics

- On-time messages
  - At the specified time.
- Bounded latency messages
  - With in a specified time interval
- Periodic messages
  - Regular telemetry data
- Order of Messages
  - because field devices do not buffer packets. Network should ensure this

- Reliability
  - Ensure packet losses are detected and reported because each message is a command to the controller.
- Safety
  - Of operation and overall system. No late (stale) packets, no duplicate packets (operate machine twice)
- Synchronization
  - Common reference for timestamps
- Security

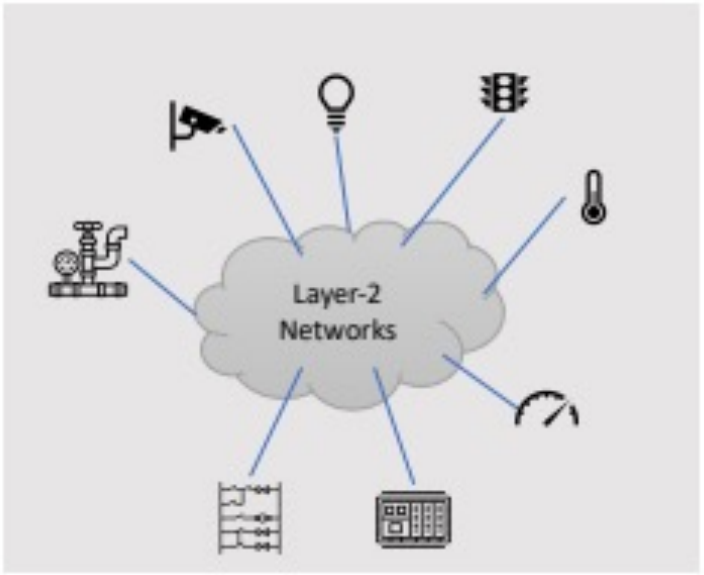
OCN Realization



Layer 1 or

Direct Physical connection

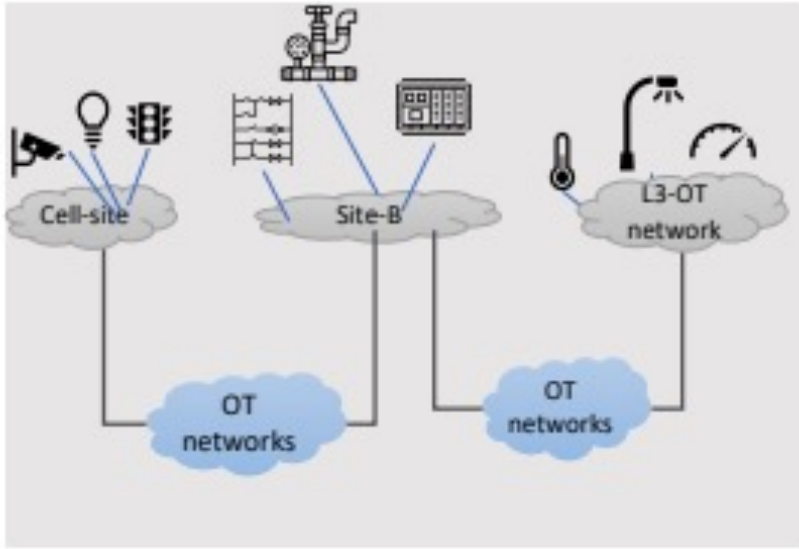
- Directly meets the high-precision requests and all types of messages
- Latency/Packet loss is NOT a concern
- E.g. Modbus, Profibus, etc.



Layer 2 or

LAN connection

- Meets high-precision requests
- Support some types of messages
- Concerns on Latency/Packet loss
- Some known algorithms & methods
- E.g., Wifi, TSN, RT-Ethernet, 5G-NR, etc



Layer 3 or

WAN connection

- Barely meet high-precision requests
- No support for OCN message types
- Concerns about Latency/Packet loss
- Some mechanisms are available
- E.g., DETNET

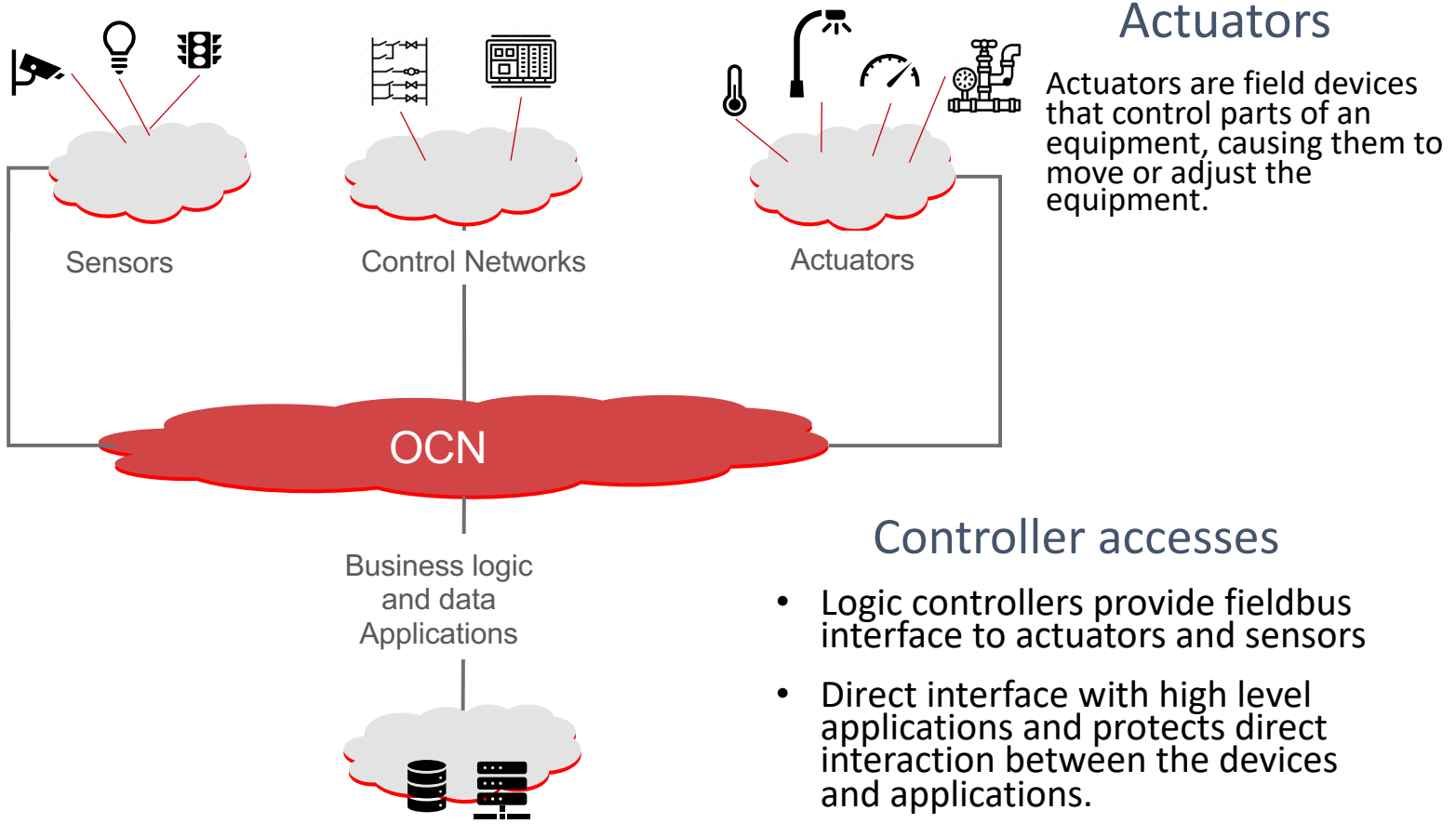
6

What is OCN?

Types of access interfaces involved are field-bus serial protocols such as Profinet, BACNet, Modbus, etc. Many sensors are connected via wireless interfaces.

Sensors

- Sensors are field devices that read values related to a machinery (pressure, position, torsion), or environment (room temperature, humidity, etc.).



Actuators

Actuators are field devices that control parts of an equipment, causing them to move or adjust the equipment.

Application Networks

- Enterprise applications that interface with controllers for data collection and customization of processes.

Controller accesses

- Logic controllers provide fieldbus interface to actuators and sensors
- Direct interface with high level applications and protects direct interaction between the devices and applications.

Thanks!

Feedback at cedric.westphal@futurewei.com

OCN Realization

1. Enabling Protocols and Technologies

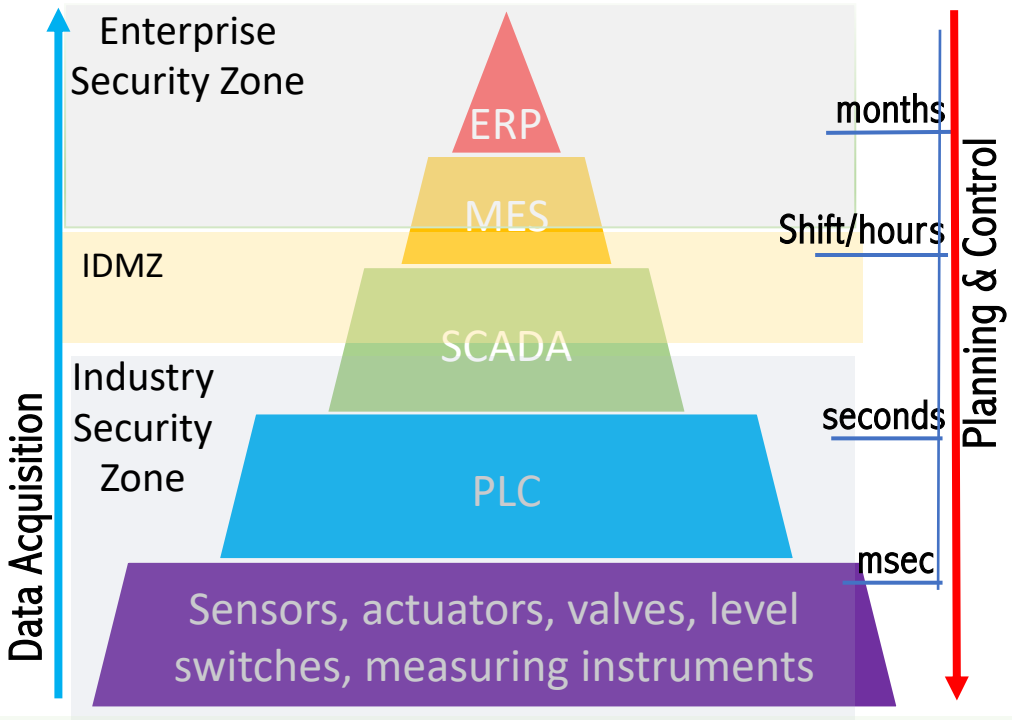
Outline

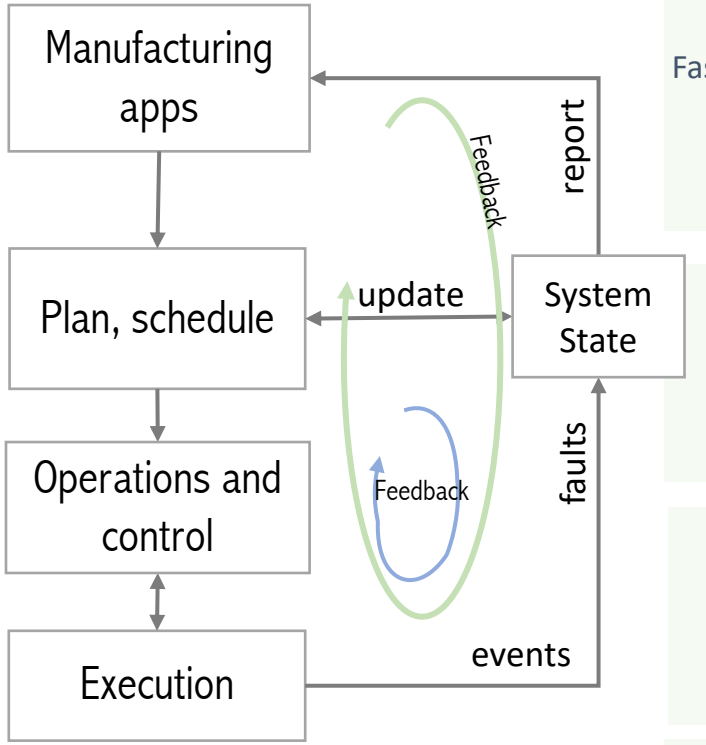
- Terminology
- IIoT and protocol limitations due to its evolutionary road
- Discuss Stacks on IIoT, IoT, OPC-UA IP leading to OCN-stack.
  - Transport - CoAP, MQTT, SCHC, TINTIN.
  - Network - IP, New-IP shipping-spec, contracts, IPv6-HBH
  - Link layer: Ethernet, 802.15.4, LPWAN
- Features: high-precision services (Network-2030), queuing, reliability, security
- Enabling technologies
  - Req: how to represent commands – not just about time-engineering
  - TSN and DetNet
  - Talk about queuing algorithms (DetNet), and our own LBF
  - Role of provisioning, admission control.
- Sandboxing OCN - Demonstration on Linux

Terminology

- **Control**: The ability to monitor and adjust a process to give a desired output.
- **Operation**: A process of a practical or mechanical nature in some form of work or production, a procedure, manner or functioning.
- **Automation**: Mechanisms that enable machine-to-machine communication by use of technologies that enable automatic control and operation of devices and processes leading to minimizing human intervention.
- **Control loops** are part of process control systems in which desired process response is provided as input to the controller, which performs the corresponding action (using actuators) and reads the output values. Since no error correction is performed, these are called open control loops.
- **Operational Technology (OT):** Programmable systems or devices that interact with the physical environment or manage devices in that environment.
- **Operation and Control Networks**: Networks that support communication aspects of process control and operation between the two end points. The OCNs are not limited to Industry operations – such as manufacturing, factory. But apply to any kind of machine-to-machine communication. An OCN is inter-connection of field devices (actuators, sensors) and their associated controllers for the exchange of data to cause and monitor changes to the end-equipment.
- **Data Ecosystem** is a set of data gathering and analysis infrastructure, systems, or applications used in an organization.

Purdue Model - Architectural Limitations





Timescale Challenges:
Faster, manufacturing and customization cycles
Data Acquisition is slow – can't react fast enough

Operations Challenges:
Hierarchy prevents adapting to new infrastructures leveraging virtualization and compute intensive

Security Design Challenges:
Constrained movement of data prevents direct application to operations feedback.
Difficult to close cloud-edge-device continuum
Air-gapped model does not work

Multi-Stakeholder Challenges:
Indirect access to data collected, i.e., process to involve 3rd parties are extremely complex and slow.

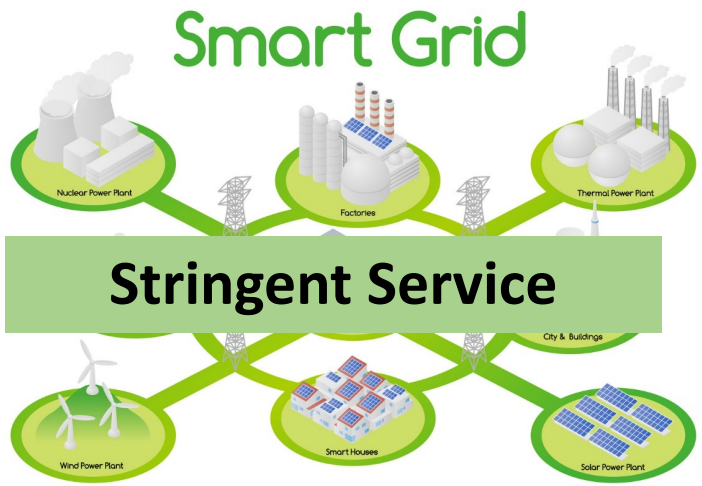
Adopts Hierarchical Approach
3 separate zones, OT and IT zones separated by IDMZ.
Data will first get collected, batched and then sent to Enterprise zone.

Challenges that drive OCN Model and framework

Breaking Down Requirements using OCN Model

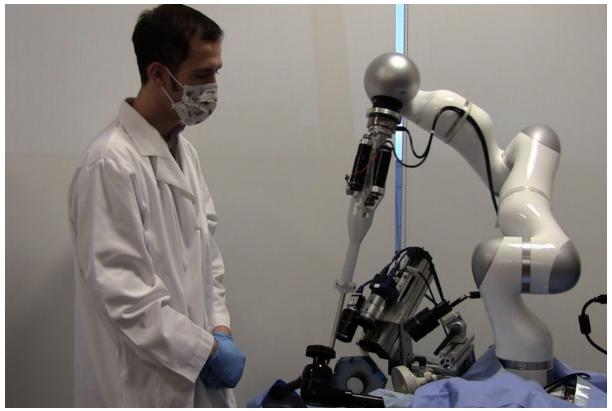
OCNs apply to a variety of Industry Verticals

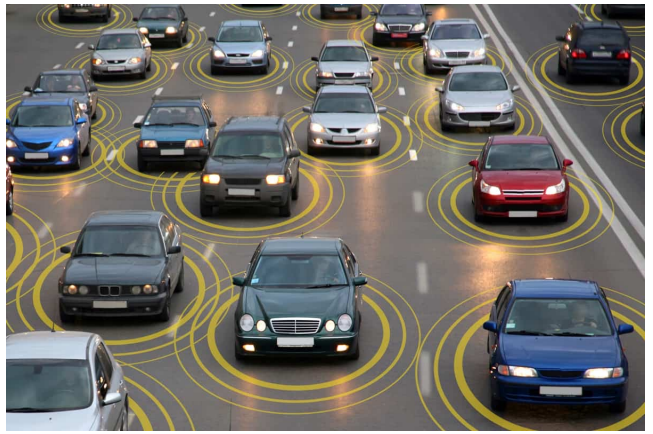
- Differentiated Services
- **Massive** number of devices (e.g., IoT)
- Stringent Services are **not always required**



Service differences from general-purpose to control systems

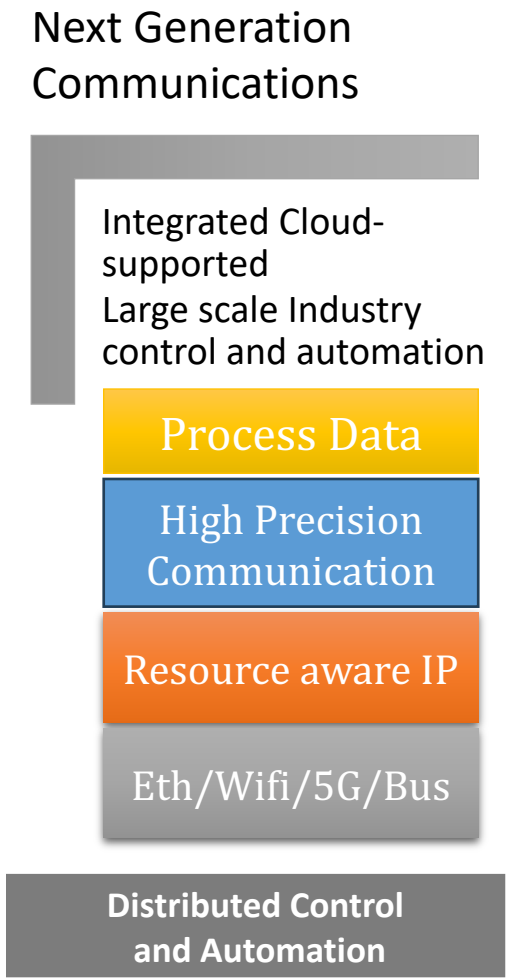
- **Best-Effort Connection** (e.g., checking non-urgent email)
- **Bounded-Latency Connection** (e.g., car braking)
- **Hard-Service Connection** (e.g., autonomous surgery)





Understanding Gaps for Next-Gen Protocols

- 4th industrial revolution is inflection point for networking and its protocols
  - Process data
    - Connections are not long-lived.
    - OCN Devices are resource constrained full IP headers tend to be too expensive
  - Resource Sensitive IP:
    - Extended Data-centric ecosystem need software to control the environment – in a timely manner and reliably.
    - Edge and Cloud infrastructures required for data computations and analysis.
    - Disaggregated composition of networks for multiple stakeholders.
- Protocol Evolution:
  - Support broad range verticals: manufacturing, transportation, healthcare, retail
  - Not re-invent the wheel unnecessarily –provide capabilities for high-precision communications.



FUTUREWEI
Technologies

OCN Realization

2. Technology Stacks and Protocols du jour

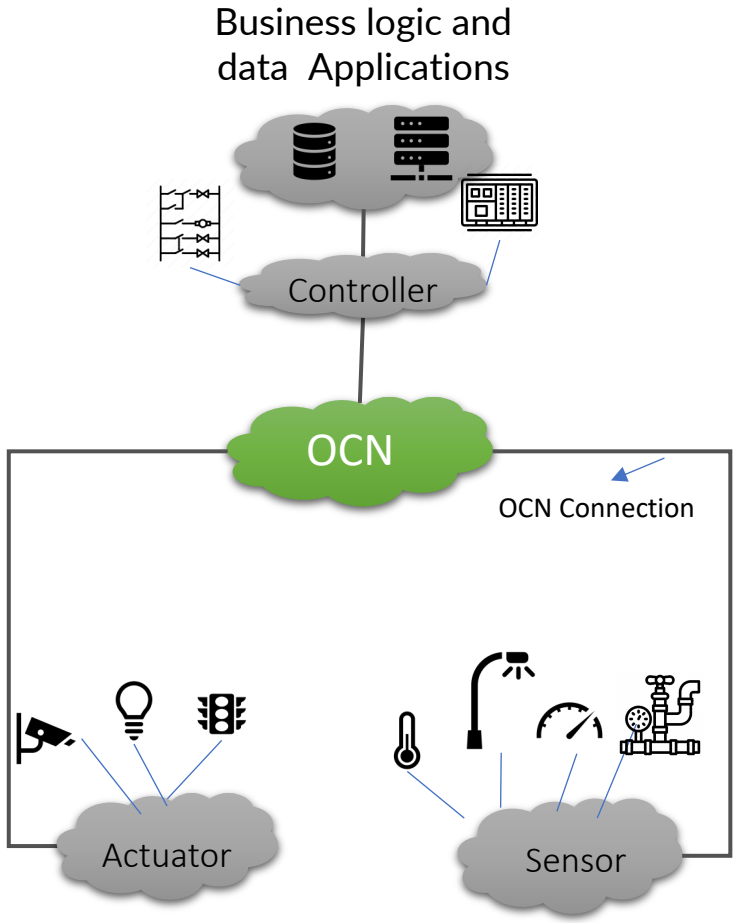
Recap OCN

Motivation

1. Growth in number and complexity of “remote operations” type use cases in which network are central to field devices and the controller communications.
2. Demands a reduced, abstracted model that multiple stakeholders can use to develop understanding about the system.
3. OCN abstracts communication primitives and interfaces in a technology neutral manner.

New asks for remote-operations & automation

1. **A scalable solution** - auto-configured, programmable, application centric.
2. **Cloud-native approach**: abstraction, elastic and on-demand scale in/out, based on virtualization and self-defined network techniques. Application behavior does not change where they get placed.
3. **Inter-operable**: Standard interfaces, to support multi-platform, multi-domain environments.
4. **High-precision communication operations**: Preserves all the characteristics of Operational Technology (OT).
5. **Compute/AI and ML:** Provide infrastructure to consume OT data.



Realizing OCN - Challenges with IP

- **IP protocol and therefore network is unaware of identity**
  - Ethernet and IP addresses very difficult and indirect way to recognize devices
  - (DNS)-name assignments non-automatic / role-driven
  - Hard to express requirements: “Firewall Policies”, “quality of service”, “reliability”, “Interconnections across gateways”
- **Identifying traffic purpose by TCP/UDP port numbers is difficult to impossible**
  - Especially due to end-to-end encryption (what traffic uses TCP port 80 / 443 ?)
  - Encryption leads to network operations are harder to understand, diagnose, secure and support traffic riding across it.
  - Accurate performance diagnostics: Why was throughput lower than desired ? Was problem in network or on IoT device (TCP stack)
- **Complex solution composition due to large number of “gateways” integrating transport traffic proxying to overcome network limitations**
  - E.g.: Preassigned static 10.x.y.z addresses for industrial devices -> complex orchestration of NAT gateway functions
  - Firewall, analytics (aggregation), control flows.
  - Connectivity needs to be cheap, but huge variety of one-off solutions today causes gateway low reliability and higher cost

Realizing OCN – Even More Challenges with IP

High Precision Data Delivery

- Different types of commands require different latency delivery
- Need network components to handle the complexity of communication characteristics such as latency, reliability, packet loss.
- Varying characteristics such as bounded latency, periodicity, message urgency on per command basis
- Packet loss avoidance and/or faster detection – can lead to a missed command to an appliance

Computational Abilities

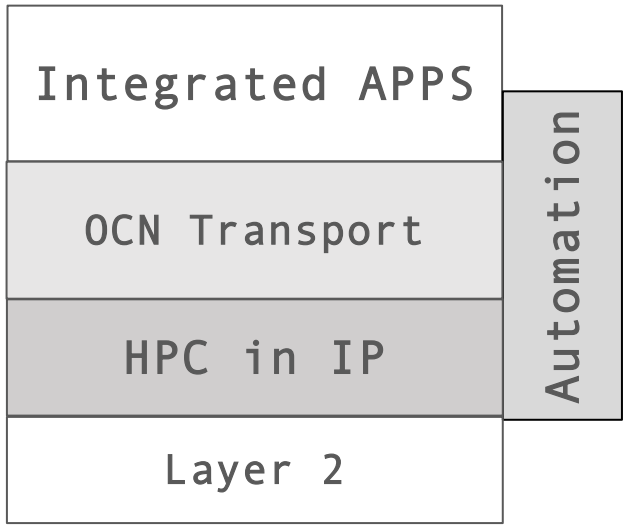
- Complex Process automation programs require computational abilities which are absent in sensors etc.

Use of AI/ML Technologies

- Predicting maintenance windows, general wear and tear of equipment requires feeding real-time data to ML models -- Connectivity to offsite server farms with such capabilities

OCN Represents Converged IIoT* and IT**

Simplify Stack



Stack derived from expected OCN Capabilities

Reduce Data Translation across Layers

- Abstractions create layers and layers cause overheads.
- Direct impact on the degree of precision you want to achieve

OCN Traffic Centric Network

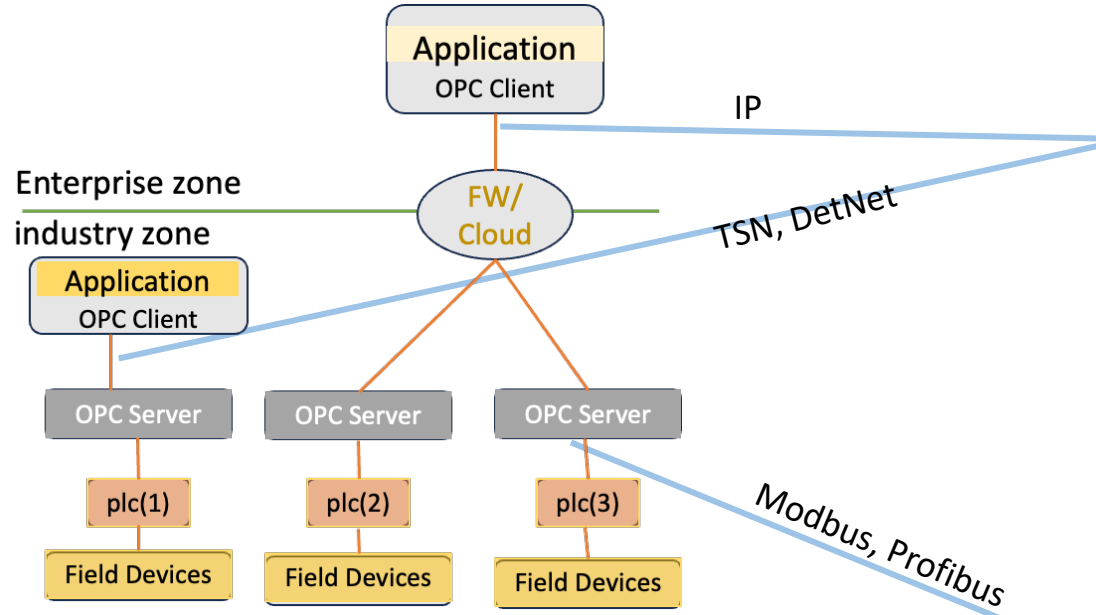
- Understand coexistence of traffic patterns
- Provide means to express goals (e.g., closed control loop, periodicity, command-based E2E, etc.)

Dynamic/Unified Interface to OCN requirements

- Different latency guarantees
- Guarantee reliability, scheduling, shaping.

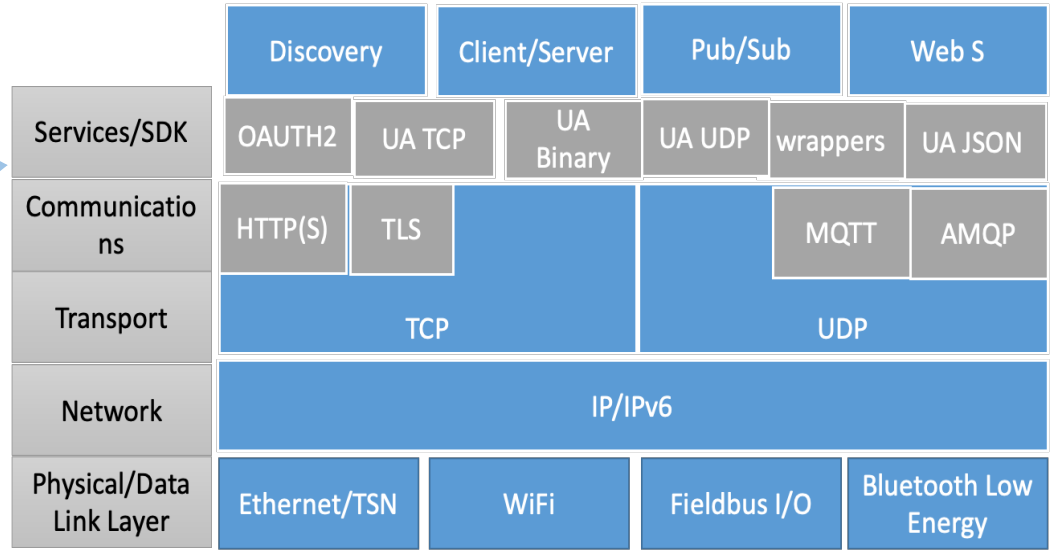
[*] Process Control
[**] Enterprise applications

Contemporary Industrial IoT Stack



Open Platform Communication Unified Architecture (OPC-UA)

- OPC Server: interfaces with the PLCs, vendor provided.
- OPC Client: collects or performs data exchange with server.

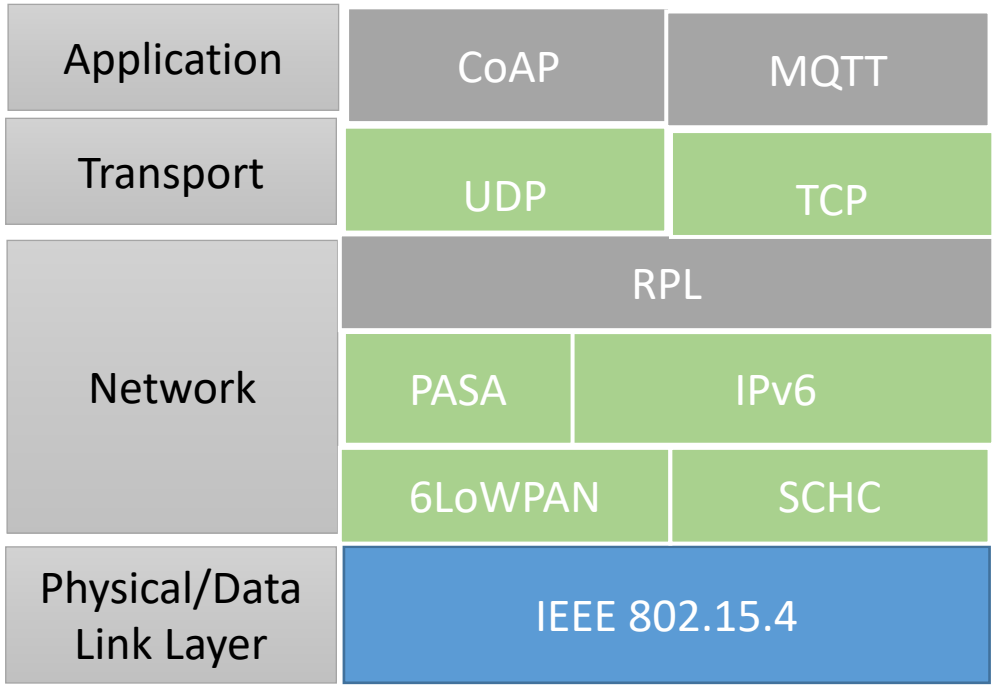


- Fundamental Protocols are Fieldbus and TSN.
- OPC standardizes manufacturer-independent data exchange.
- Limitations in field level protocols remain the same (no E2E High Precision Communication)

this is a stack for data collection, does not do device control well.

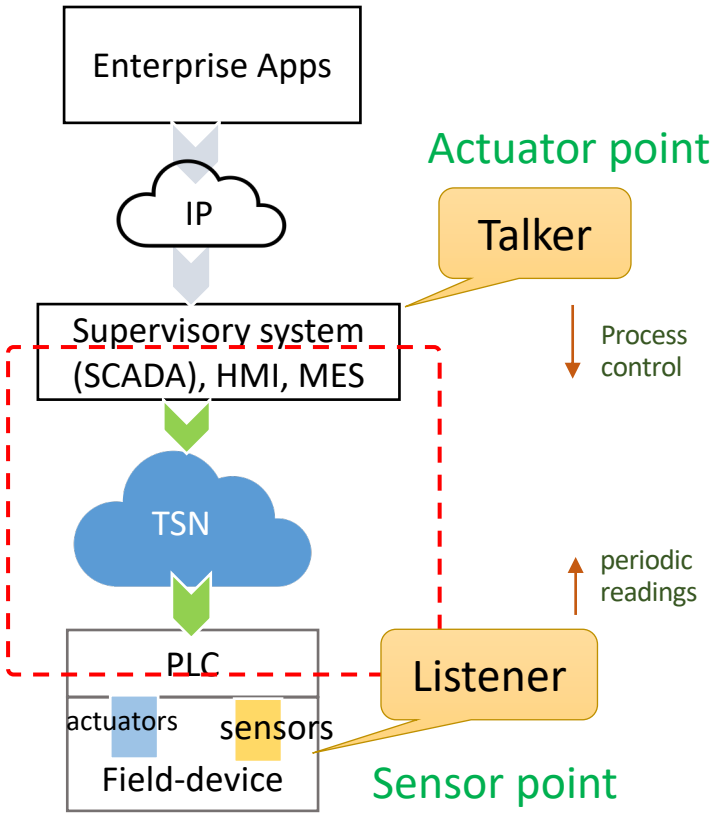
Source: https://www.opc-router.com/what-is-opc-ua/

Contemporary Architecture: IoT Stack



- IETF has developed several protocols for resource constrained IoT devices.
- Mainly applicable to radio based PHY media.
- overlaps across protocols some may become relevant to Industrial IoT.

Time Sensitive Networks as OCNs



- In depth work has been done to develop shapers, queueing algorithms.
- Careful admission control: The configurations are performed network-wide, i.e., each TSN bridge must be configured with the profiles provided through the central controller (or control plane distributed).

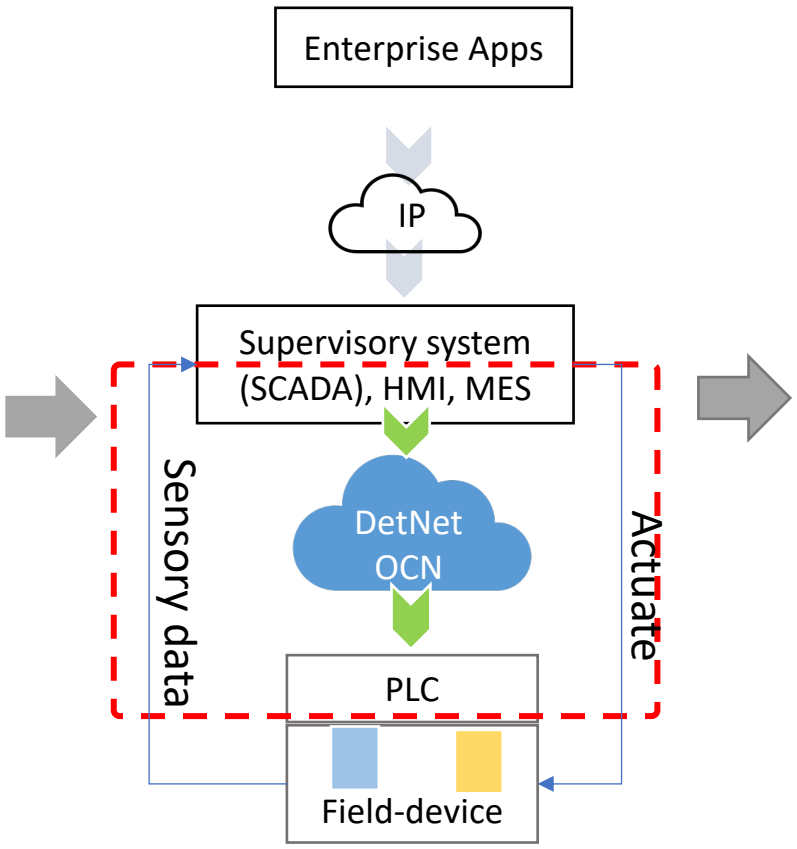
TSN Tool	Engineering Complexity	Wire Efficiency	Worst Case Latencies – 1st Order Approximation	Ranking
Time Aware Shaper	Hard (>1 TC) Medium (1 TC)	- Guard band - Idle opens	15.7 uSec per FE Hop 2.0 uSec per GE Hop	2
Frame Preemption	Medium but only 1 level deep	- Fragment overhead	27.5 uSec per FE Hop 3.2 uSec per GE Hop	3
Credit Based Shaper	Easy	100%	249 uSec per FE Hop 138 uSec per GE Hop	1
Strict Priority	Easy	100%	Easy	Not Applicable

Concerns

- TSN lacks programmability and thus application related customizations.
- Application diversity and interface to network is limited (operator-defined profiles instead)
- Scalability over large networks – response to dynamically changing conditions.

Source: https://www.itu.int/en/ITU-T/Workshops-and-Seminars/20180127/Documents/3.%20Norman%20Finn.pdf
https://www.allaboutcircuits.com/industry-articles/choosing-the-right-tsn-tools-to-meet-a-bounded-latency/

Deterministic Networks and OCN



- Primary goal was to interconnect TSN and it is evolving now to a standalone technology. Scalability over large networks is addressed.
- Has developed and architecture based on service layer and transport layer. Supports ISP view of operating DetNets.

DetNet Tool/feature	Engineering complexity	Wire Efficiency	IETF status and effort
Transport/service sublayer	Needs to configured on Detnet relay nodes	Bound to protocols	RFC
IP data plane	Relies on DSCP markings		
MPLS data plane	MPLS networks may not be suitable for IIoT networks	efficient	RFC
Scheduling mechanisms	Recently started discussions	-tbd-	Work in progress
Application I/f	YANG model based (svc prov.). In-band solutions are	none	I.D

Concerns

- Also lacks programmability and thus application related customizations.
- Application diversity and interface to network is limited (operator-defined YANG models instead)
- Response to dynamically changing conditions

OCN Parameters mapped to DetNet Sublayer

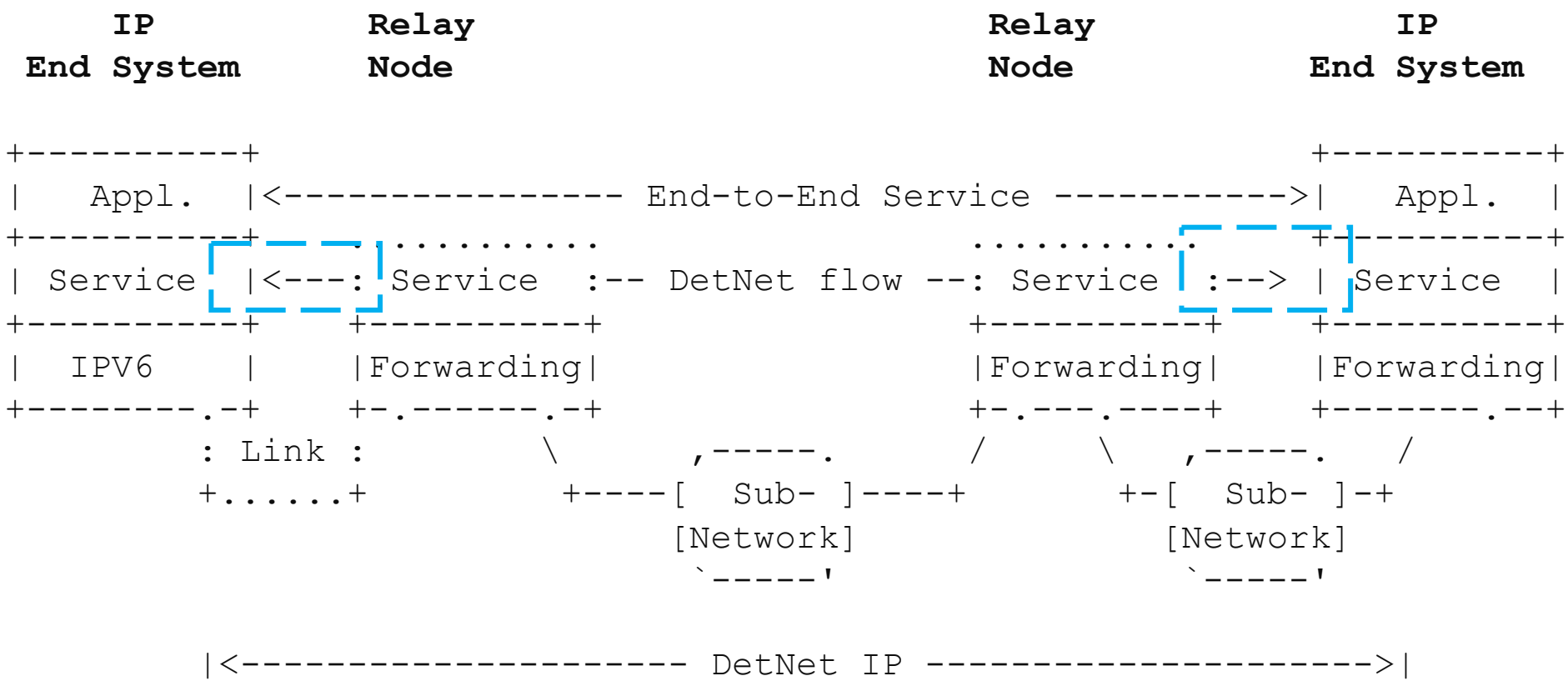


Figure: A Simple DetNet-Enabled IP Network, Ref. RFC8939

OCN Option (OCNO) as an EH option

- Motivated by HBH enhancements [draft-ietf-6man-hbh-processing-06] Nodes should not drop HBH packets if they don't process them.
- Forward looking cloud-based systems will support IPv6

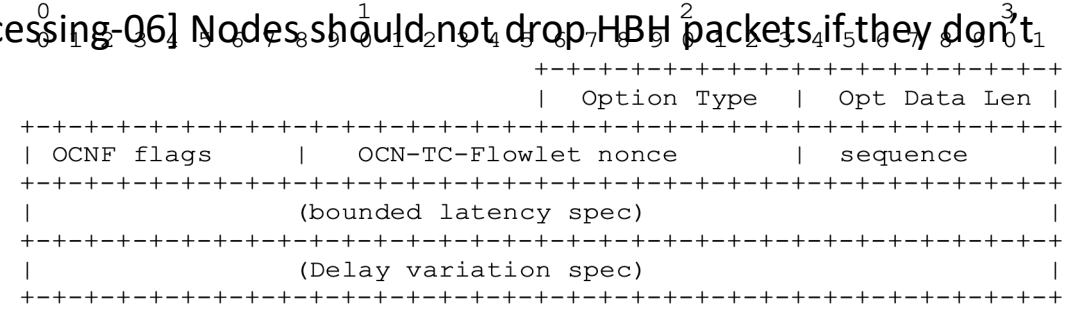


Figure 4: Explicit Traffic Control HBH Options

Flag	Description
U	send message immediately. its an alarm
P	periodic packet (intervals in ~ms)
F	part of flowlet. see Nonce and seq
L	bounded latency spec provided
R	Reliability with no packet loss tolerance
V	Delay variation with no packet loss tolerance

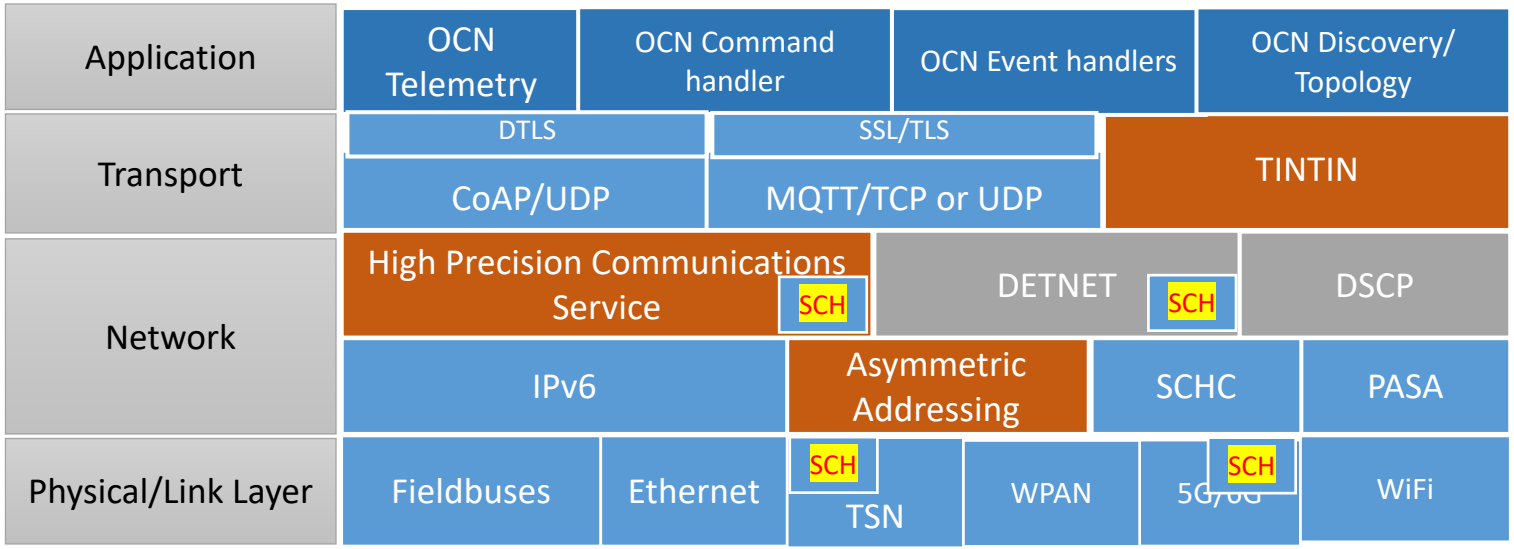
- Provides a reasonable way to interact with the DetNet Relay nodes
- Programmatic interface is application friendly (M2M Comm.)

OCN Realization

3. Emerging Technologies

Contemporary and Emerging Protocols

- Application Level or Platform
  - OPC-UA (Open Platform Communication)
  - OneM2M
  - RiOT-OS
  - OpenThread
- Transport Protocols (End-2-end communication)
  - TCP, UDP
  - CoAP, MQTT (over UDP)
  - TINTIN
- Network Layer
  - IP over TSN, DETNET (IP, MPLS), SCHC
  - New Research directions: Asymmetric Addressing, HPC Constructs,
- Link Layer
  - TSN, ProfiNet, EthernetRT, BLE, WiFi, 802.15.4



OCN Addressing

Efficient and Flexible Addressing
For cloud and low power devices

HPC/SCH

Latency Based Forwarding (Queuing Discipline)

Lightweight Transport

Secure single protocol
Carrier for commands
End-to-end responsibilities

*SCH: Scheduling algorithm

OCN and Asymmetric Addressing

Basic Idea

- Derives from OCN Model → Cloud to device continuum.
- Provide a programmable address header in which both source and destination are self-describing or pre-configured formats.
- Eliminate the need to upgrade every device with one-kind of address (IPv4 or v6 for example).

Advantages

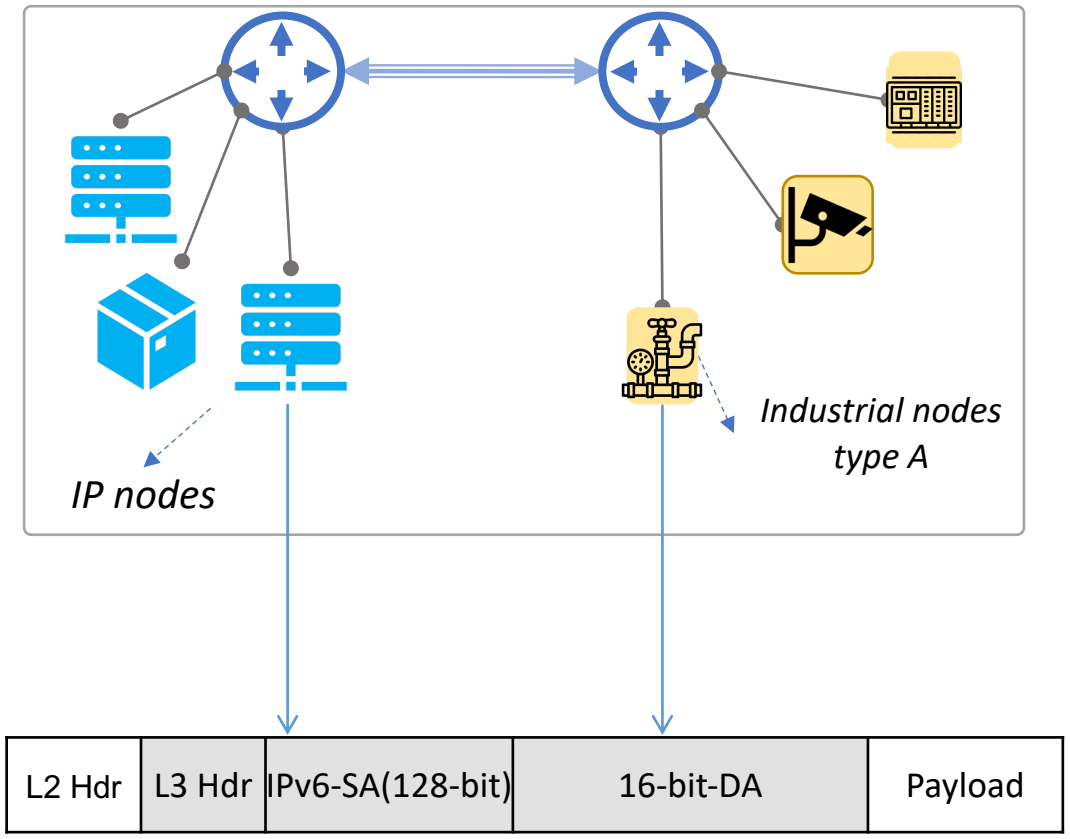
- Addresses customized for the real operational scenarios
- Routing, Discovery, maintenance is optimized.

Hierarchical (Logical)
- Device identifier
- Subnets for logical separation
- Valve-controls/floor-2

Layered or tiered (topological)
- Provides topological significance
- Tier-identifiers
- Rack/Zone/North-side

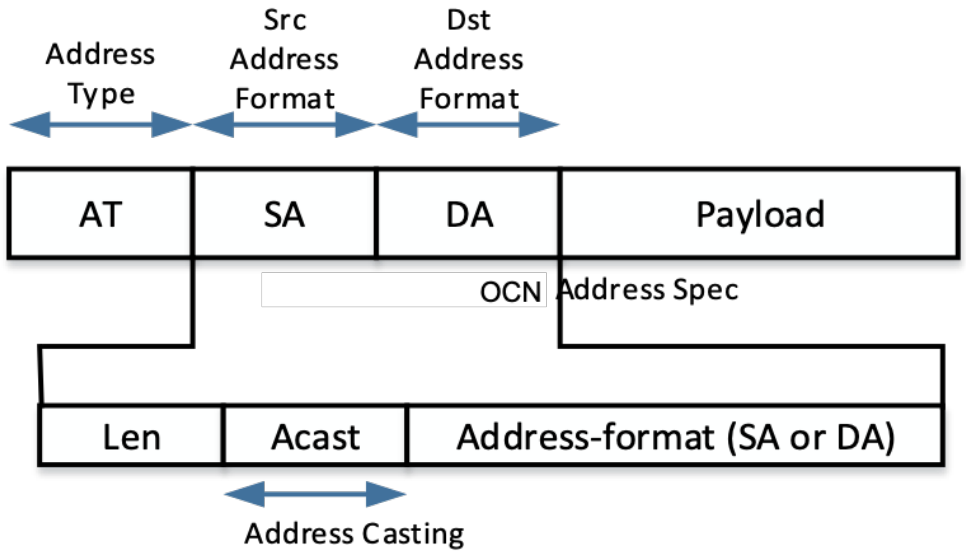
Compact
- Shorter addresses
- Modbus (8-bit), 6LoWPAN, SCHC

Semantic
- Application or service groups
- Cameras | alarms/surveillance

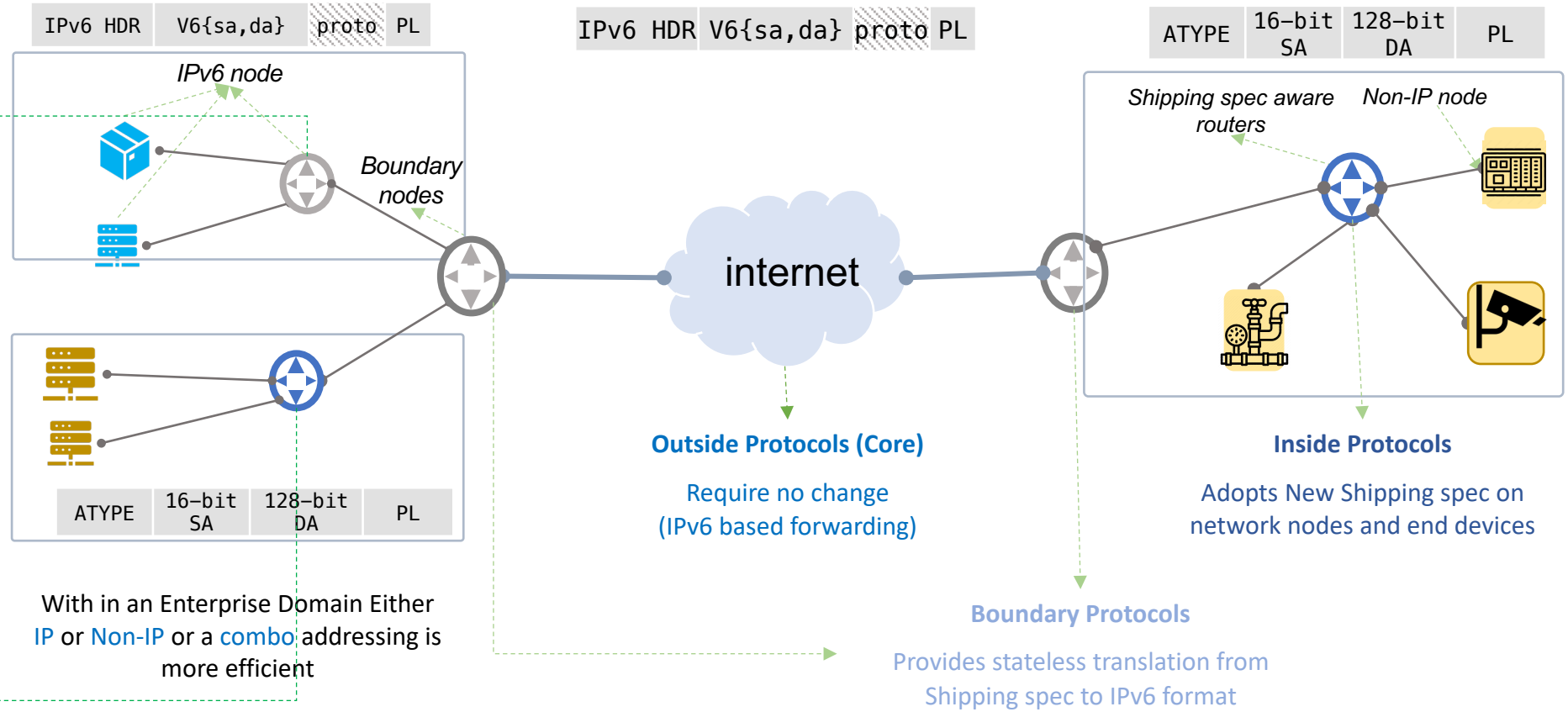


Asymmetric Address Format

- A container of the address formats supported in a system.
- Embraces asymmetry of address spaces (because addresses have different semantic meanings)
- Source and Destination address formats can be different. E.g.
  - Source can be an IP SCADA application and destination
  - Destination can be Profibus device
- Address Type is tuple to indicate the address space (or family).
  - First and second parts of the type tells source and destination (V4_ProfBus)



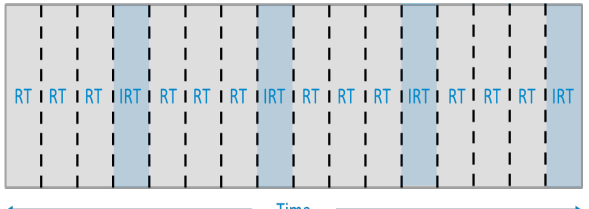
Asymmetric Address Routing and Forwarding



OCN and High Precision Communications

- **Traffic Distribution Guarantees**
  - Per packet expectations: Different types of commands require different delay budgets.
  - Varying characteristics such as bounded latency, periodicity, message urgency on per command basis
  - Packet loss avoidance and/or faster detection – this could lead to a missed command to an appliance.
  - Ordering – should be supported at network layer
  - Congestion Control –managed by network design (avoidance at end-points)
- **Accountability Guarantees –** Sufficient and Performant collection of errors
- **Cyber-Security Threat Protection**
  - Against different threats such as taking over the role of sensor to send incorrect readings, taking control over an actuator to execute different behavior or jamming the network with burst of sensor data.
- **Safety and Reliability Guarantees** -- only correct data should be sent by devices

Towards High Precision Communications

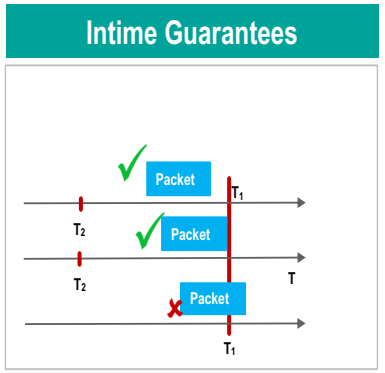


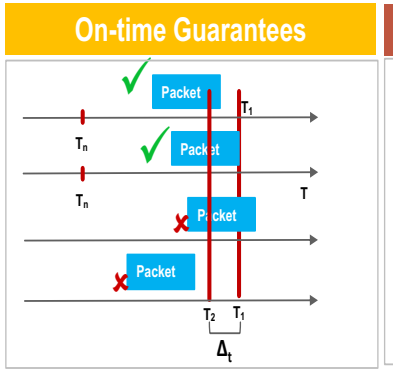
Source: https://www.motioncontroltips.com/what-is-isochronous-real-time-irt-data-exchange/

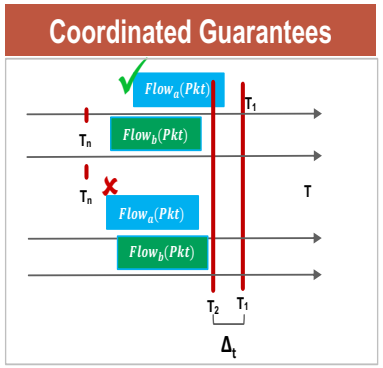
Assign Time slices for isochronous real-time (IRT) and standard real-time (RT) communications.

Clock Synchronization

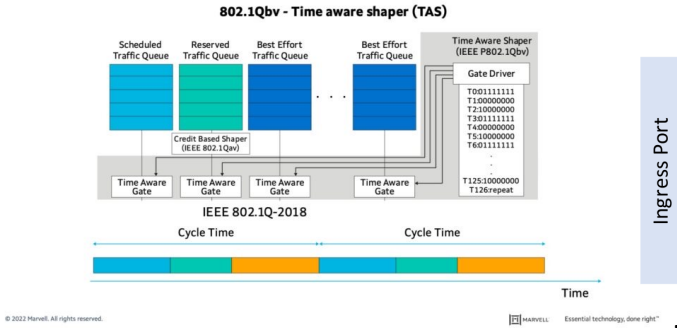
Declarative Latency Requests





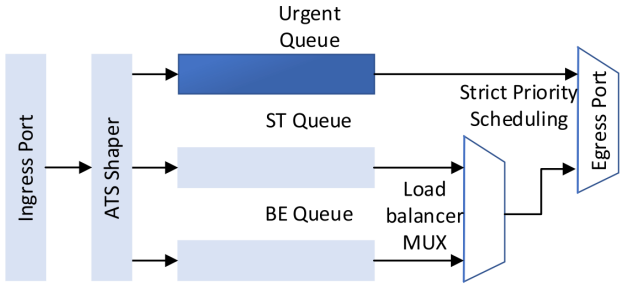


Low/bounded **latency**



Time Aware Shaper

Clock Synchronization



The ATS shaper

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8681083&tag=1

Time-based determinism/guarantees are achieved in many ways through - Profinet/TSN –RT, Non-RT, isochronous

Enabling Technologies for HPC in OCN

Method	Description
• Asynchronous Transmission Scheduling (ATS) [P802.1Qcr]	• Per-hop shaping without clock synchronization. • Maintains
• Scheduled Traffic [802.1Qbv] • Cyclic queuing and forwarding[802.1Qch]	• Periodic repeated schedule, reduces packet delay variation. • Needs clock synchronization
• Frame Replication and Elimination for Reliability (FRER) [802.1CB]	• Enhances reliability
• Delay based forwarding • Tagged Cyclic Queue Forwarding (TCQF) • Asynchronous Deterministic Networking Framework	• More queuing approaches. Some are more complex to implement in hardware than others. • Similar to TSN but being done with IP based networks in IETF DETNET to support for large scale networks.

Latency Based Forwarding (LBF)

Design

- Delivers packets within bounded latency (both `minimum` and `maximum`)
- Path Considerations
  - Uses number of hops to determine how much time the packet should wait at a node
- `Experienced_delay` stored/modified en-route in the packet
  - no need of clock synchronization
- LBF contract (generalized `metadata` structure) packet includes
  - Minimum delay, Maximum delay, experienced delay, fib_todelay, fib_tohops

Operation

- Ideally, the packet should spend number of hops/minimum delay (fib_tohops/min_delay) at each hop
- Time to be spent at each hop = (minimum delay - delay experienced)/number of hops left

- If experienced delay greater than maximum delay then drop the packet
- Packet is dropped if the number of hops exceed the expected number of hops

Limitations:

A. Hop count

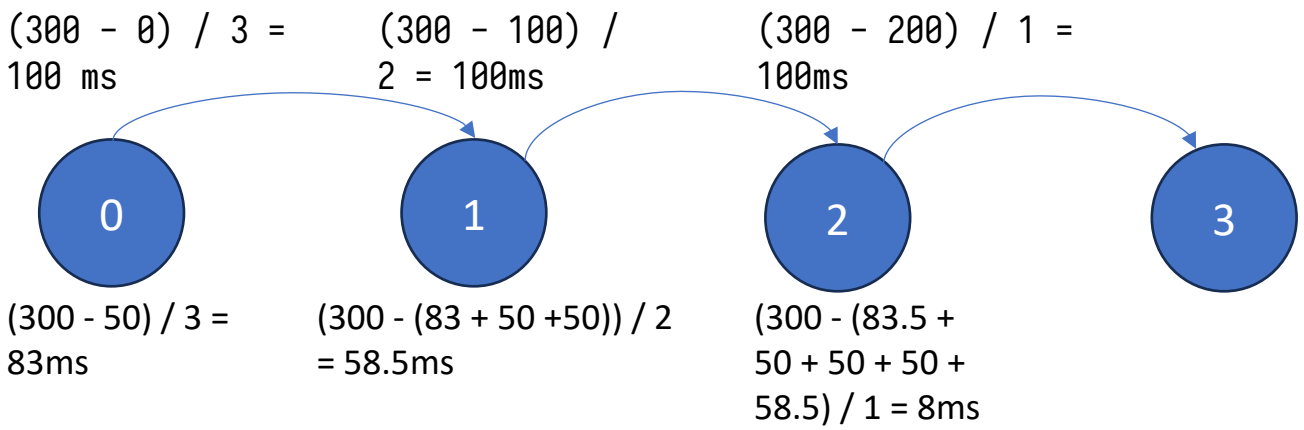
B. Experience Delay modified.

LBF Examples

At any node:
packet delayed by = (minimum delay – experienced delay)/number of hops

4 nodes (0, 1, 2, 3)
number of hops = 3
minimum delay = 300ms

Sender:
no queueing delay
experienced-delay = 0



Sender:
no queueing delay
experienced-delay = 0

OCN and TinTin

Tiny In-Network Transport for High Precision INdustrial Communication)

Although less constrained than IoT, the communication in Industrial IoT is characterized with the following:

- Time-critical
  - Every operation is executed in or at a specific time precisely without complex state management
- Safety-critical and Reliable
  - The accuracy of each command received is utmost critical; the data cannot be lost or arrive incorrectly
- Resource-constrained
  - It is extremely important to maintain a light-weight network stack on field-devices
- Stateless and Session-free
  - Neither session based QoS is useful nor maintenance of long-lived sessions on endpoints

TinTin Design Principles

- Features suitable for industrial networks (OCN aligned)
  - Supports long and short protocol control headers
  - Closed-control loop
  - Publish/Subscribe messages
  - Time-centric packet delivery
- Assumes a High Precision Communication Network is available
  - For time-critical end-to-end delivery.
- No port numbers
  - Uses a ‘magic token’ instead
  - TinTin control header introduces a ‘message-type’ concept

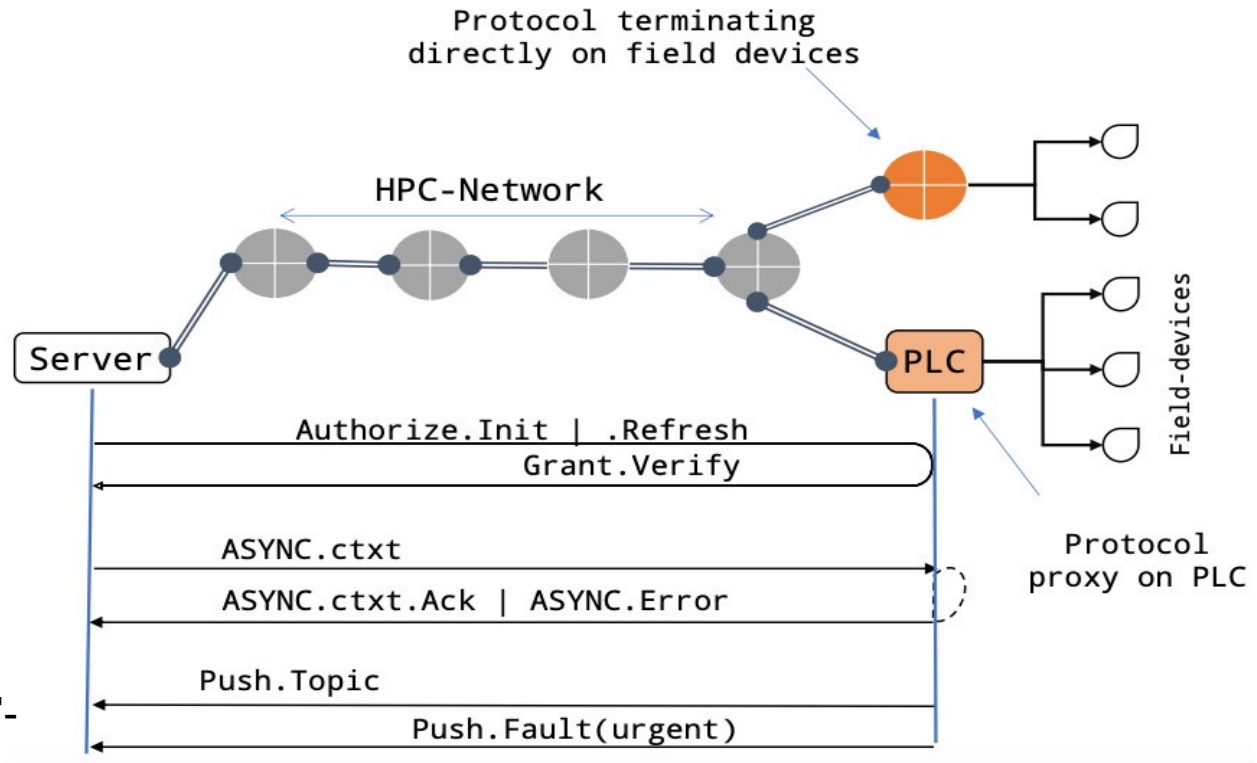
Key considerations of End-to-End Guarantees TinTin protocol:

- Packet Delivery: connectionless protocol with application driven reliability, modular choice of long/short headers
- Enhanced Reliability: due to additional time based parameters provided by TinTin, does not send an ACK unless requested
- Congestion Control: industrial networks are well managed, so TinTin relies on the network to handle congestion or expects pacing at the server side
- Selective Packet Ordering: notion of a flow is not necessary in IIoT, most of the messages contain short commands, sequence numbers are provided to maintain ‘order among a group of commands’.

TinTin: High-Level Architecture

Terminology:

- TinTin Endpoint: field-devices, application servers or controllers
- Authorized node: that is verified and granted access to by a field device
- TinTin Protocol Data Unit (T-PDU): the transport control header of TinTin
- Payload: service or application data in T-PDU



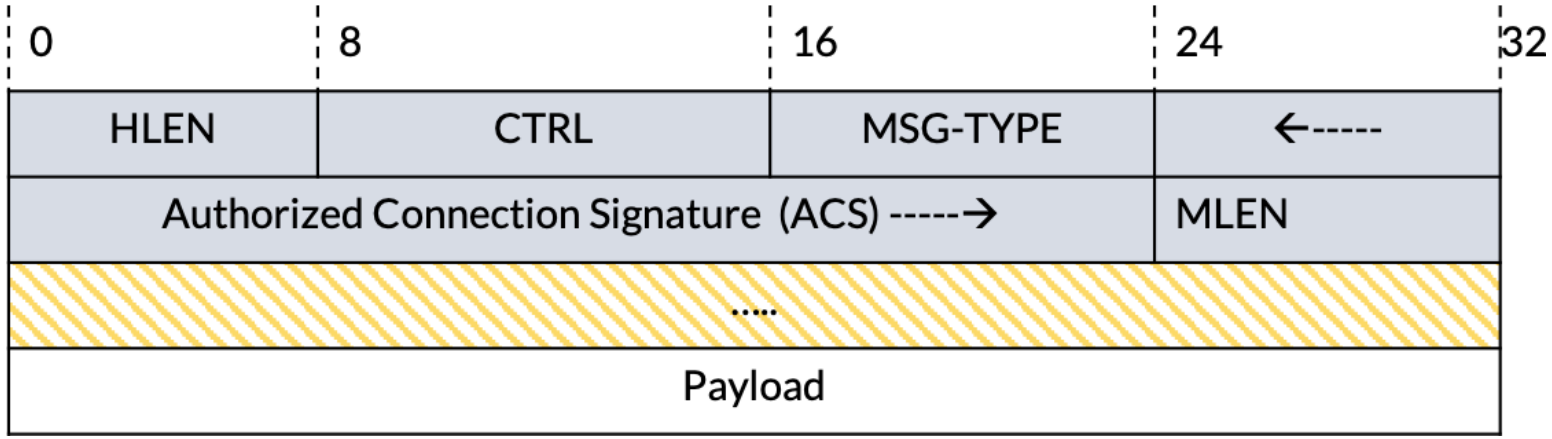
TinTin Architecture and Communication Model

Protocol Operations

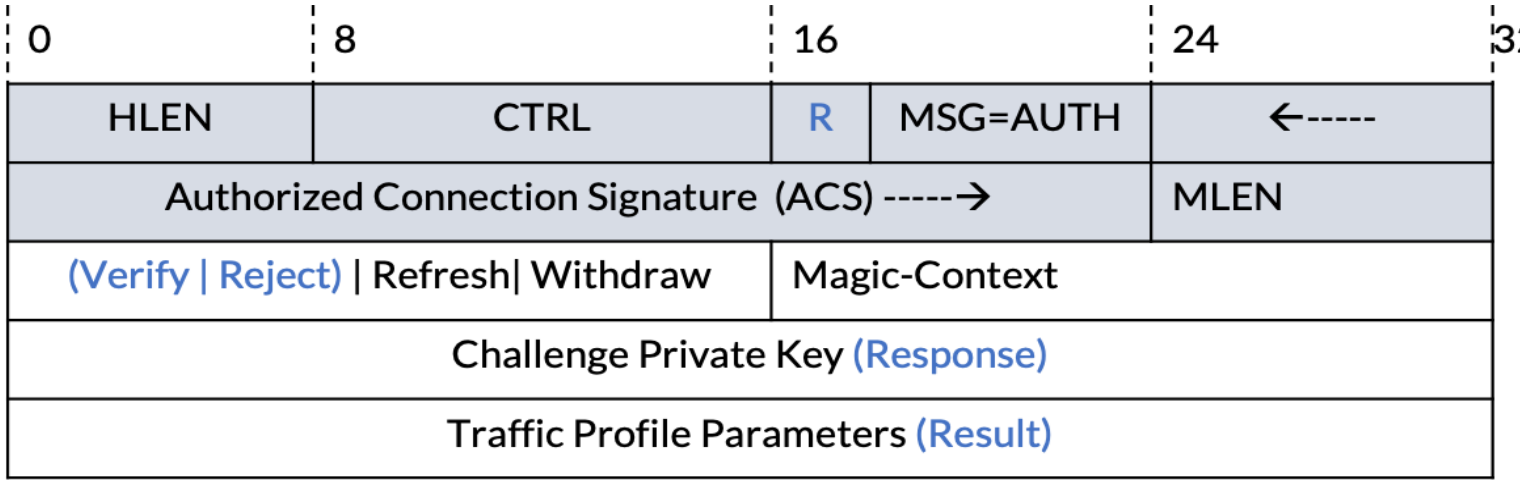
Three top-level pairs of directive(s): Minimal Header Fields:

- ❑ Authorize & Grant
- ❑ Async & Reply
- ❑ Post

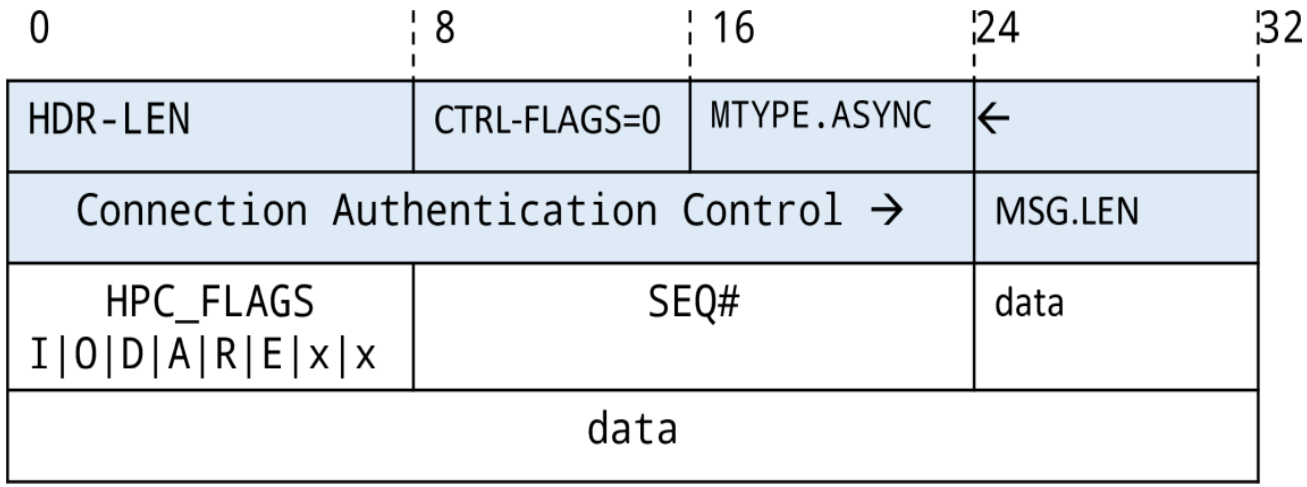
- ❑ ACS: Application Connection Signature
- ❑ CTRL: placeholder for specific extension flags or to indicate length of ACS



Directives



Authorize and Async Headers

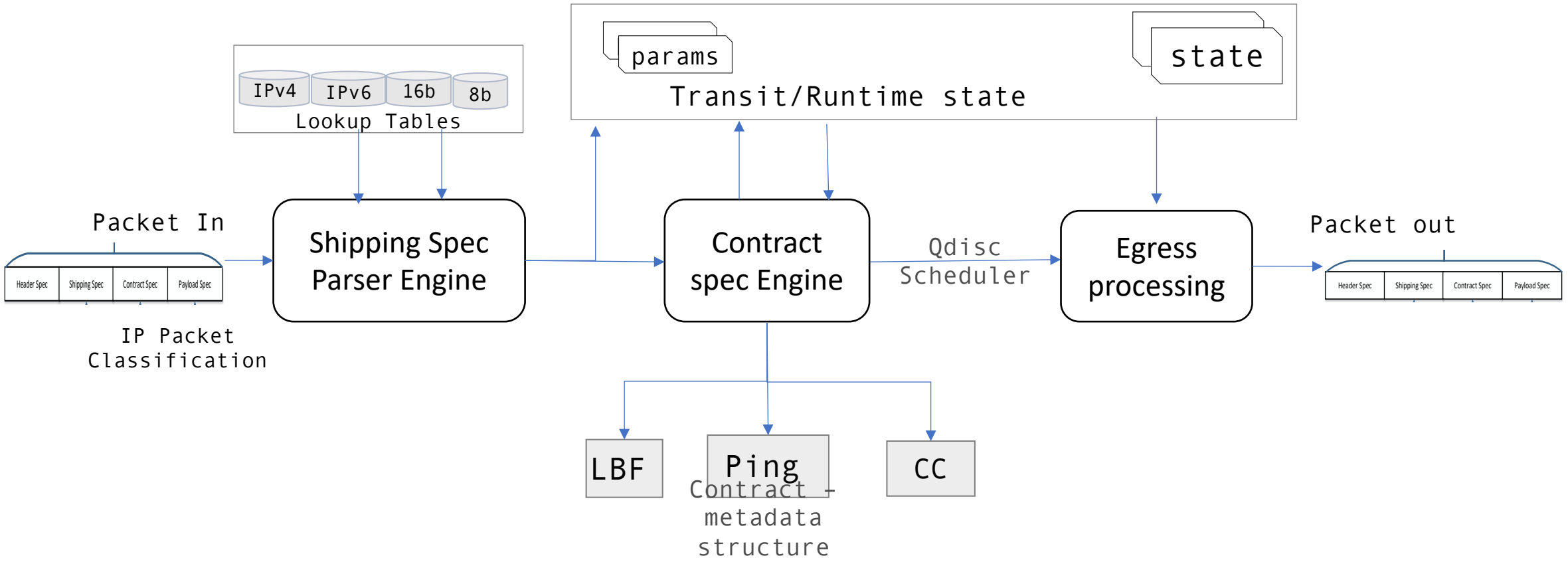


Aysnc Header for time-based transmission

OCN Realization

3. Proof Of Concept (Demo)

Conceptual OCN Packet Processing Pipeline



What is NeST: Network Stack Tester

- A Python package to emulate networks
- Uses network namespaces to simplify the process of setting up networks
- Provides intuitive APIs to
  - Build a virtual network topology
  - Run experiments on the virtual network topology
  - Collect statistics
  - Plot results

Why NeST?

- Simplifies the process to reproduce network experiments
- Less physical resources, less error prone and less prerequisites
- Multiple instances of the same network topology can co-exist, and different experiments can be run in parallel on every instance
  - NeST assigns random names to the network namespaces
  - Cleans up the environment (deletes network namespaces) on termination
- Open source tool released under GPLv2 License

Peer to peer topology

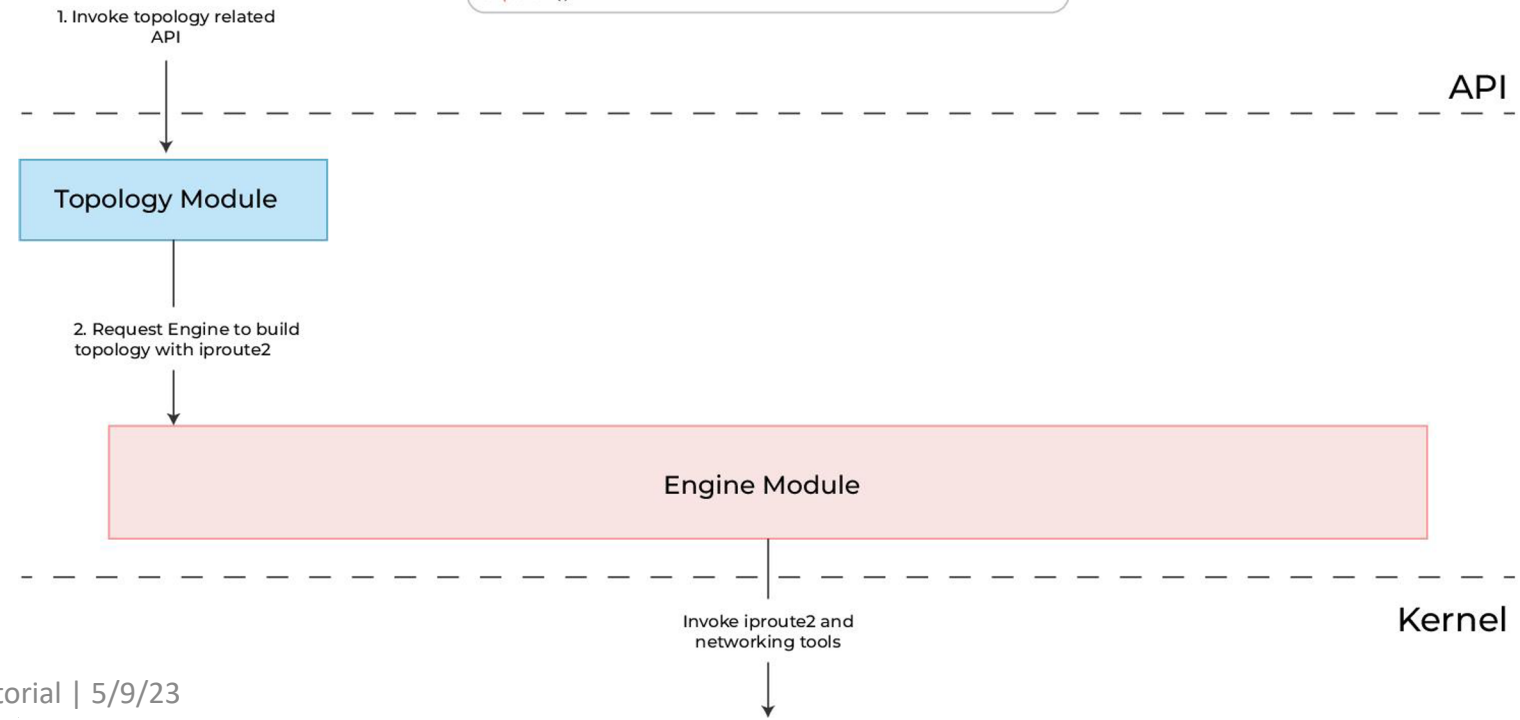
```
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')

# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)

# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')

# Set link properties
n0_n1.set_attributes('5mbit', '5ms')

# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')

# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))

# Run the experiment
exp.run()
```

Peer to peer topology

```
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')

# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)

# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')

# Set link properties
n0_n1.set_attributes('5mbit', '5ms')

# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')

# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))

# Run the experiment
exp.run()
```

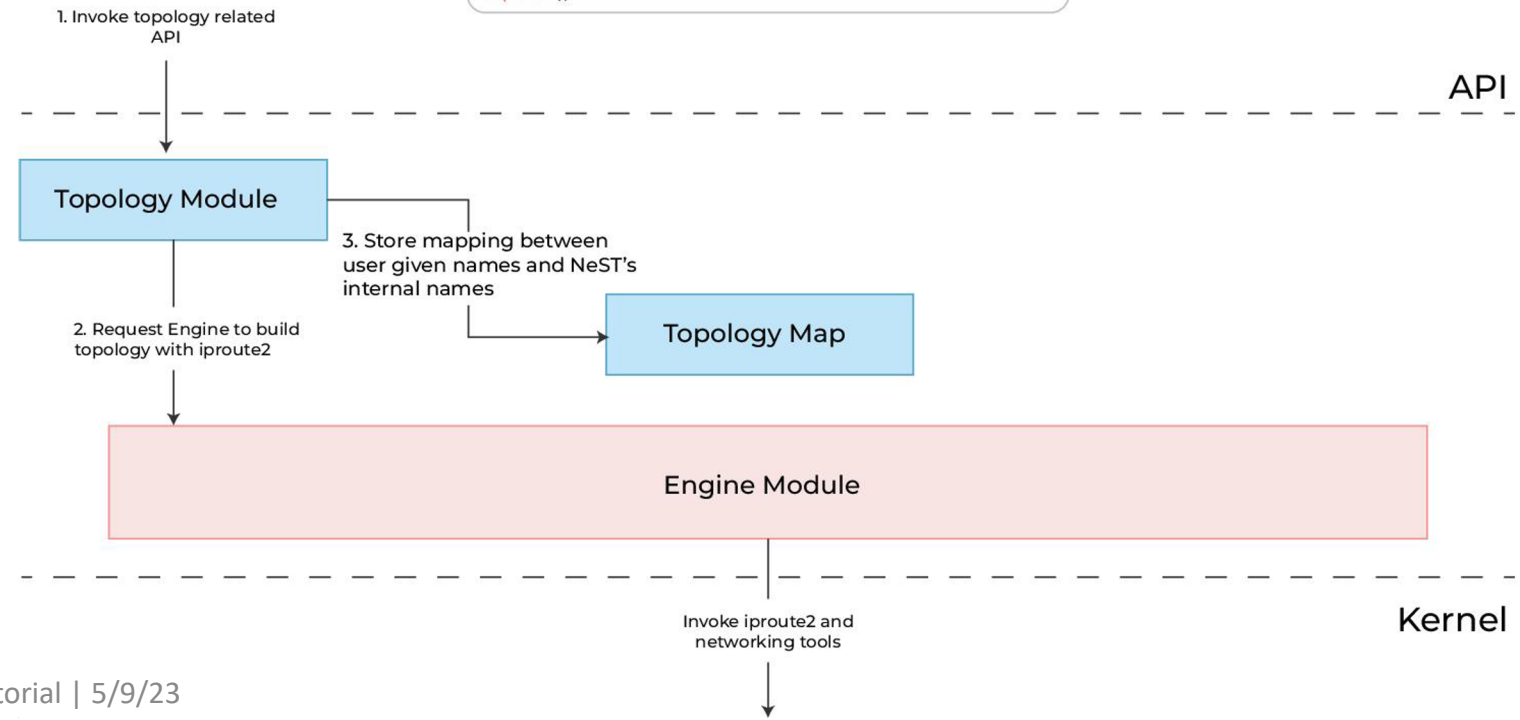
Peer to peer topology

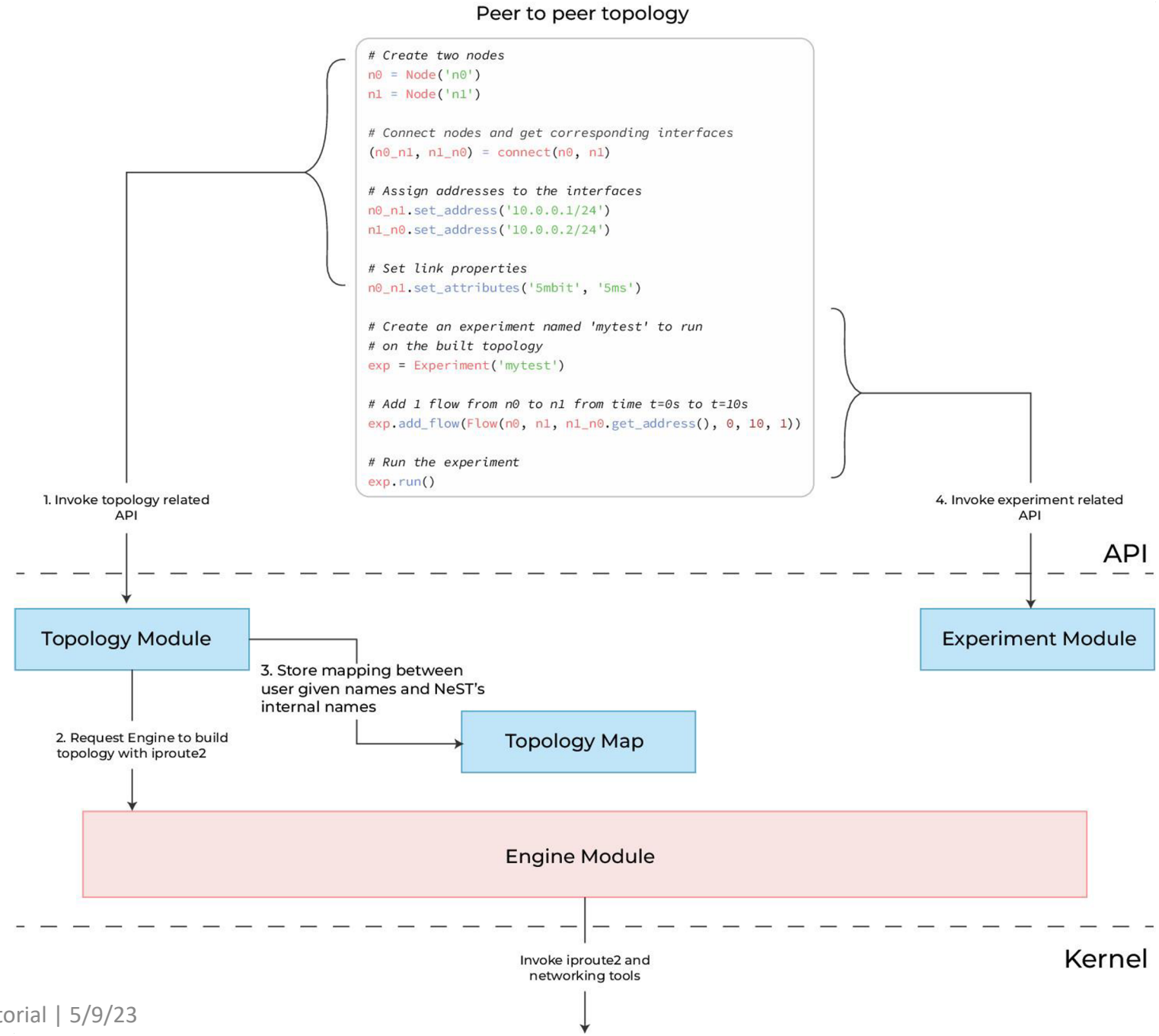
```
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')

# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)

# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')

# Set link properties
n0_n1.set_attributes('5mbit', '5ms')

# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')

# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))

# Run the experiment
exp.run()
```

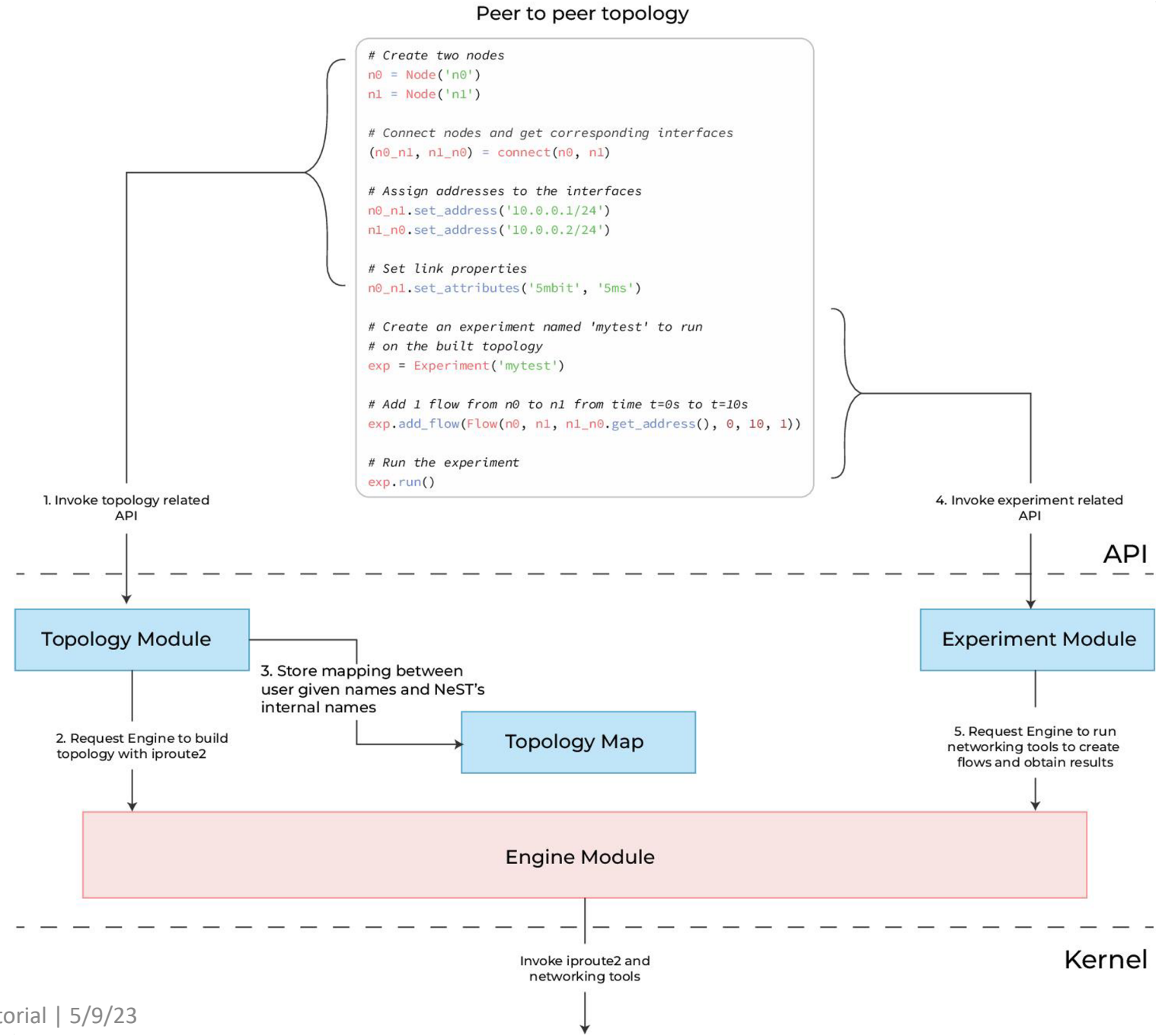


Peer to peer topology

```
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')

# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)

# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')

# Set link properties
n0_n1.set_attributes('5mbit', '5ms')

# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')

# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))

# Run the experiment
exp.run()
```

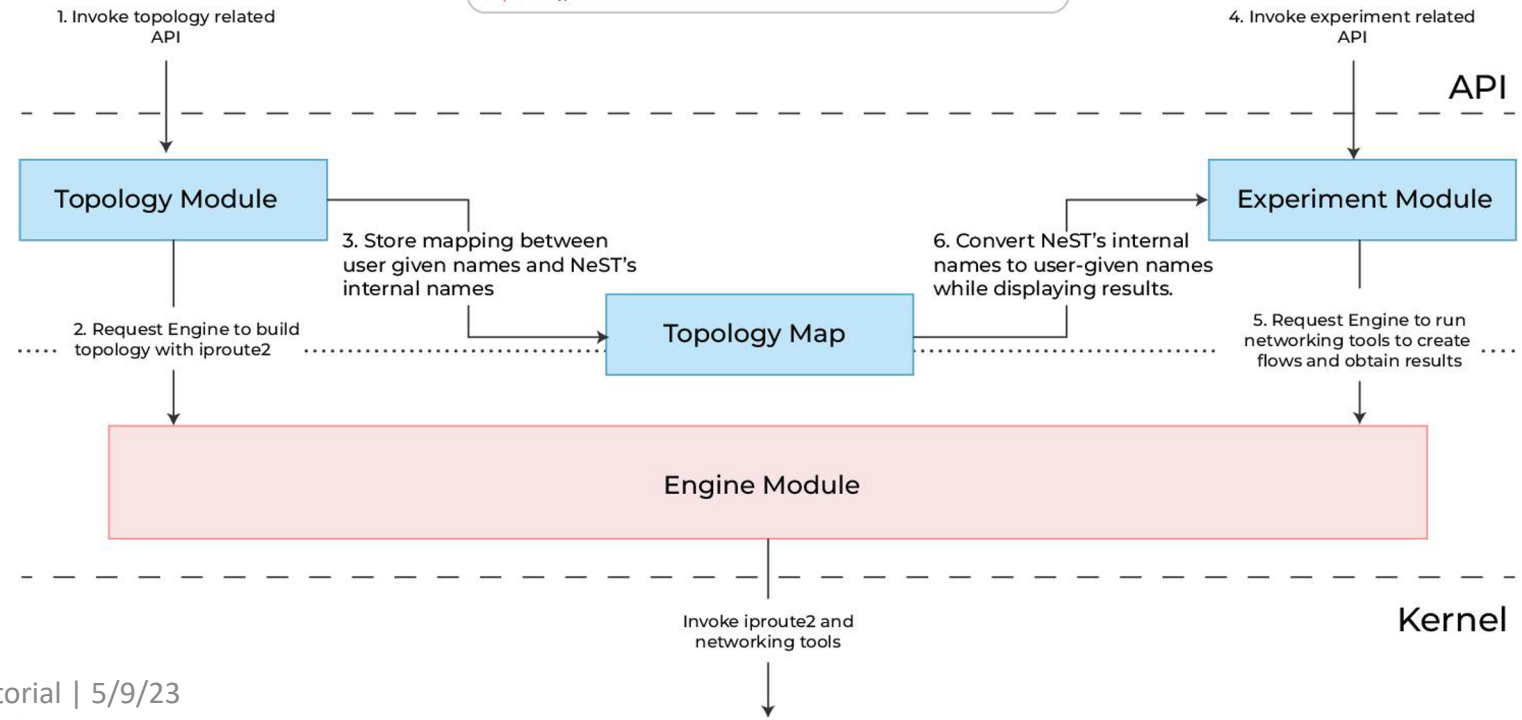



Peer to peer topology
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')
# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)
# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')
# Set link properties
n0_n1.set_attributes('5mbit', '5ms')
# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')
# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))
# Run the experiment
exp.run()
1. Invoke topology related API
4. Invoke experiment related API
API
Topology Module
Experiment Module
3. Store mapping between user given names and NeST's internal names
Topology Map
2. Request Engine to build topology with iproute2
Engine Module
Invoke iproute2 and networking tools
Kernel


Peer to peer topology
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')
# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)
# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')
# Set link properties
n0_n1.set_attributes('5mbit', '5ms')
# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')
# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))
# Run the experiment
exp.run()
1. Invoke topology related API
4. Invoke experiment related API
API
Topology Module
Experiment Module
3. Store mapping between user given names and NeST's internal names
Topology Map
2. Request Engine to build topology with iproute2
5. Request Engine to run networking tools to create flows and obtain results
Engine Module
Invoke iproute2 and networking tools
Kernel

Peer to peer topology

```
# Create two nodes
n0 = Node('n0')
n1 = Node('n1')

# Connect nodes and get corresponding interfaces
(n0_n1, n1_n0) = connect(n0, n1)

# Assign addresses to the interfaces
n0_n1.set_address('10.0.0.1/24')
n1_n0.set_address('10.0.0.2/24')

# Set link properties
n0_n1.set_attributes('5mbit', '5ms')

# Create an experiment named 'mytest' to run
# on the built topology
exp = Experiment('mytest')

# Add 1 flow from n0 to n1 from time t=0s to t=10s
exp.add_flow(Flow(n0, n1, n1_n0.get_address(), 0, 10, 1))

# Run the experiment
exp.run()
```



Linux Toolset used

Host side API: Scapy

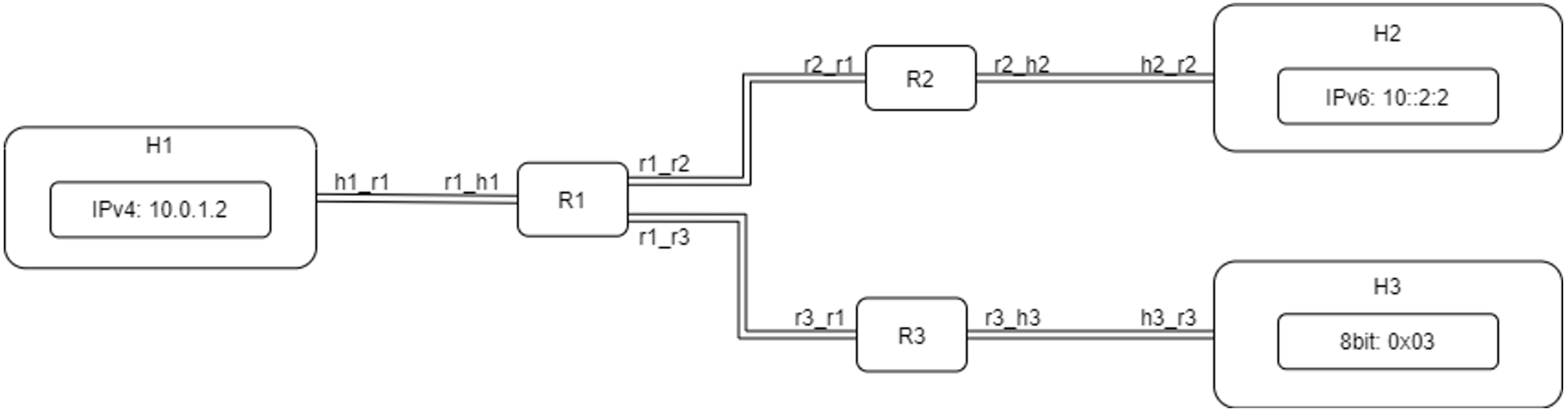
- Craft and send the packets from the virtual interface setup by NeST
- Sniff and decode the packet at the receiver for analysis

Network side eXpress Datapath (XDP) and Traffic Control (TC)

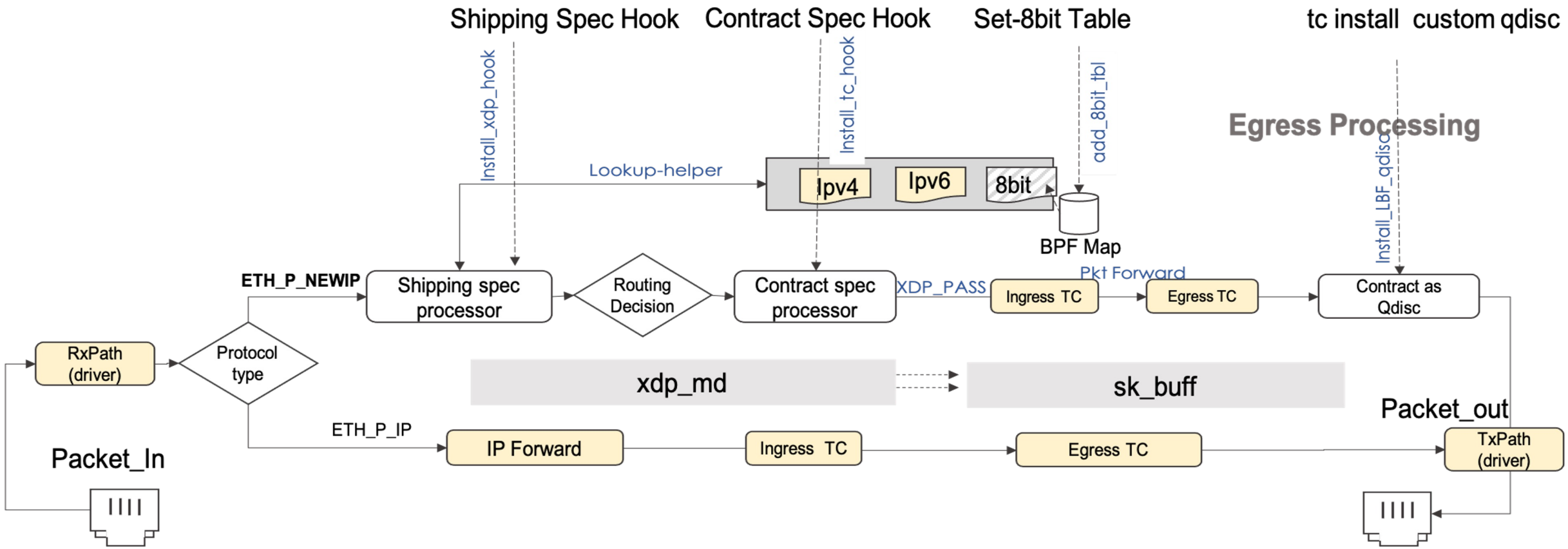
- XDP is an eBPF based high performance data path
- Provides packet processing at the lowest point in the software stack
- Identify the interface from which the packet is to be sent using kernel routing tables or BPF maps
- Store the exit interface index (ifindex) in metadata of the packet
- Pass the packet on to the TC BPF hook

- BPF programs attached to the traffic control (tc) ingress and egress hook
- Used to add our queueing discipline (we cannot have queue discipline with just XDP)
- Read the ifindex from metadata at ingress hook
- Redirect the packet to the egress hook of the interface associated with ifindex
- Run our queue discipline on the egress hook

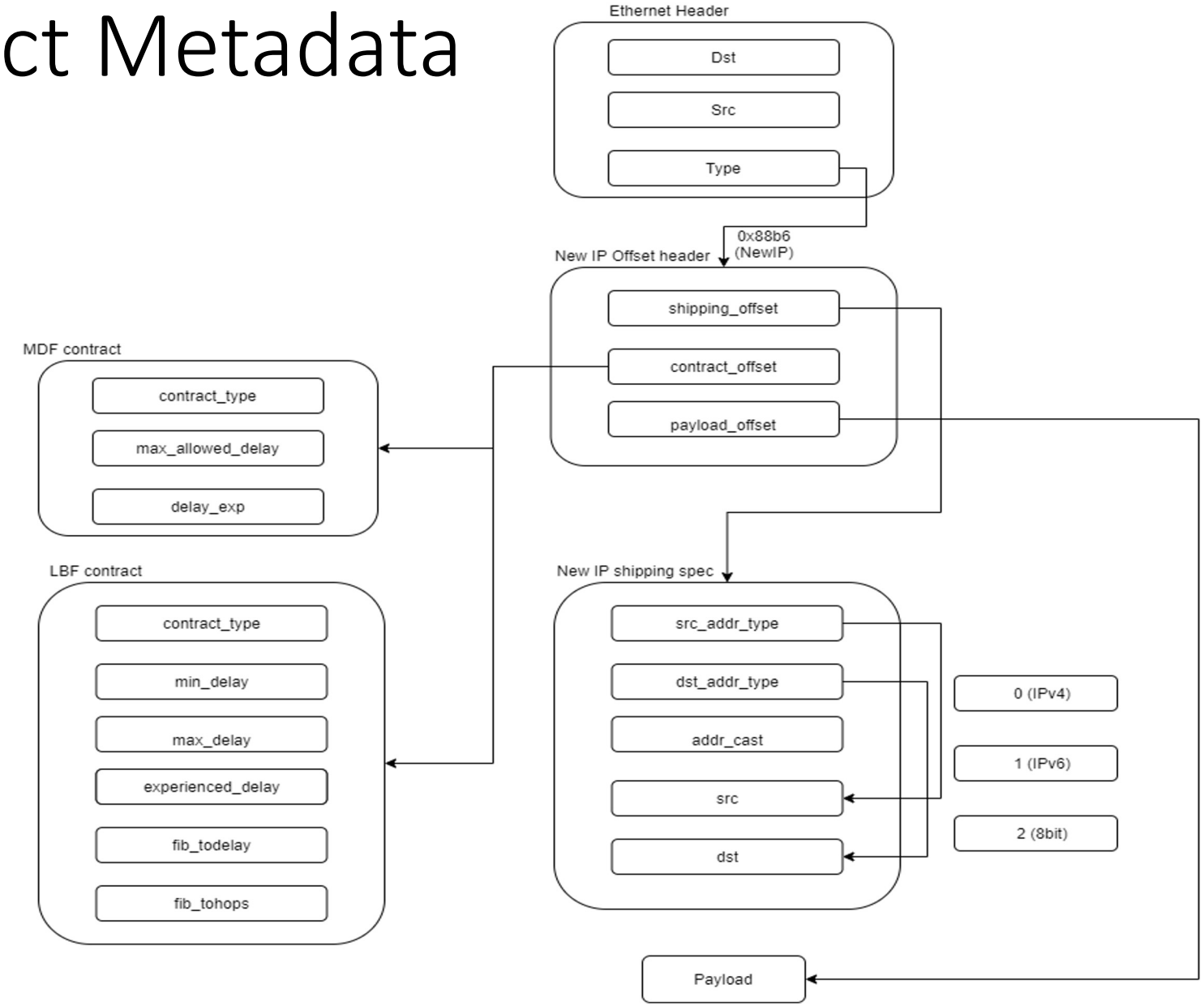
Topology Setup - NeST



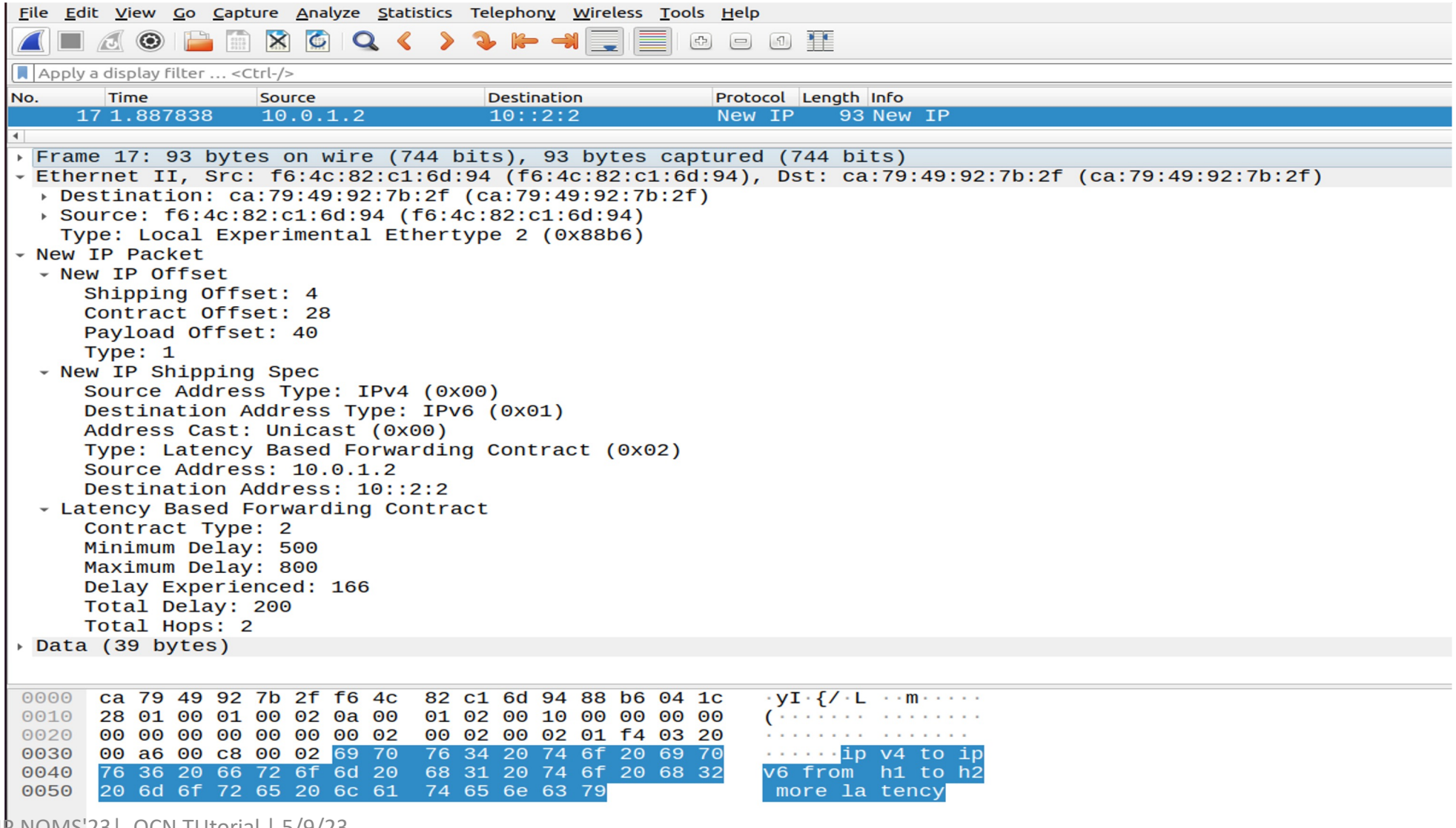
Packet processing



Contract Metadata



Packet in Wireshark



API for Crafting a New IP packet

```
setup_obj = Setup()

# setting up the Topology
setup_obj.setup_topology()

# Defining New-IP contract less flows
flow1 = LbfFlow(
    src_node=setup_obj.h1,
    dst_node=setup_obj.h3,
    src_addr_type="ipv4",
    src_addr=setup_obj.info_dict[setup_obj.h1.name]["ipv4"],
    dst_addr_type="ipv6",
    dst_addr=setup_obj.info_dict[setup_obj.h3.name]["ipv6"],
    pkt_count=10,
    min_delay=3000,
    max_delay=5000,
    hops=setup_obj.info_dict[setup_obj.h1.name]["hops"][setup_obj.h3.name],
)

# instantiating the Experiment class of NeST
exp = Experiment(name="lbf-flow with TCP and UDP  ")

# Adding New-IP contract less flows
exp.add_lbf_flow(flow1)

# Running the experiment for all added flows
exp.run()
```

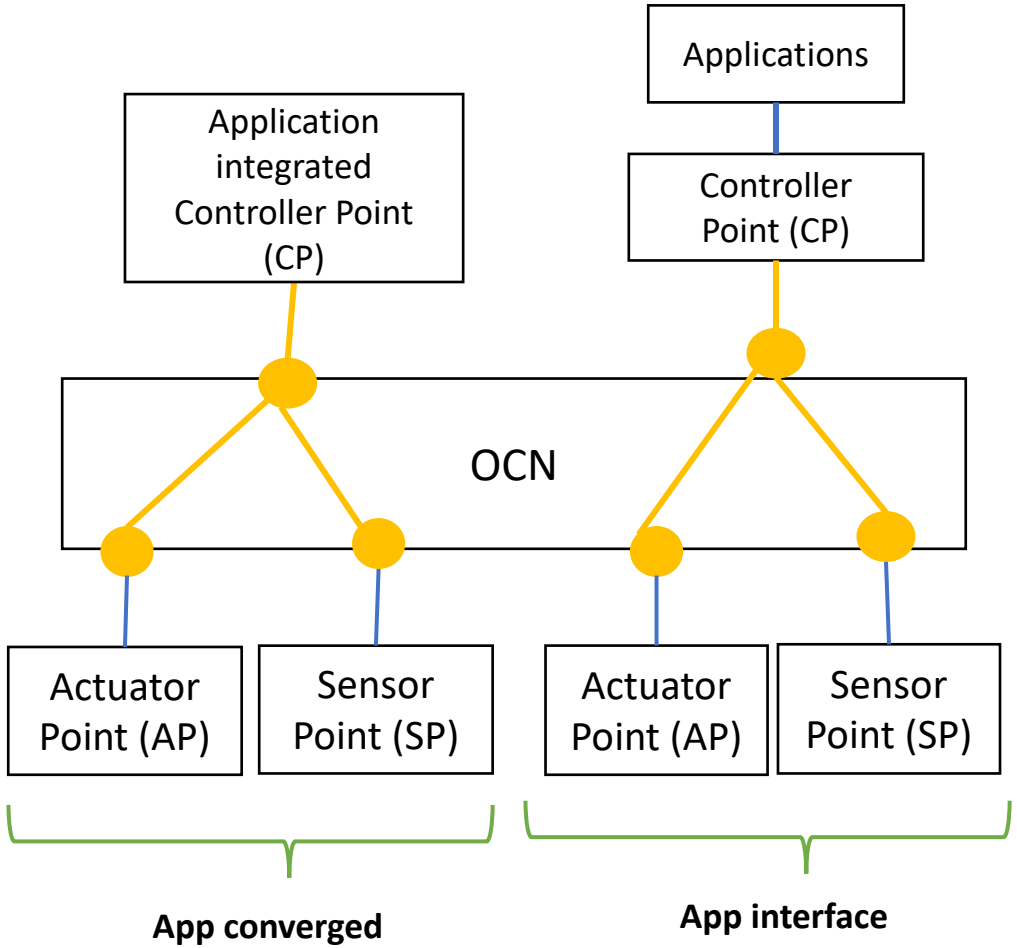
Wrap up

OCN Reference Model

OCN as an abstraction of control systems.

- Technology Agnostic

- Logical role-based Reference points
  - between: Actuators, Sensors and Controllers.
- Standardize
  - (a) interfaces and common message types, their format
  - (b) corresponding network-constraints
- Application Interface
  - Converged or independent



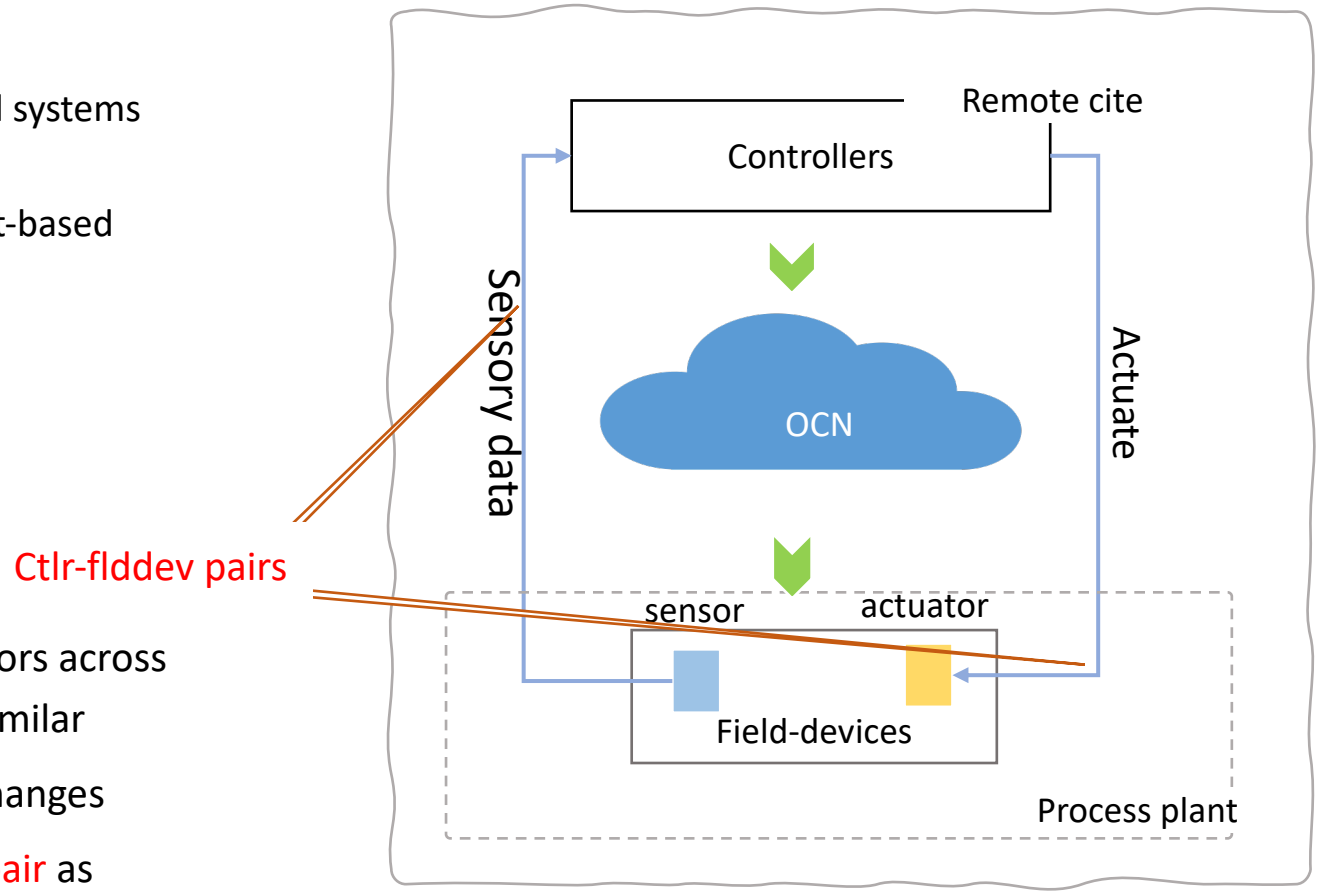
Considerations

- Operator vs Application view
  - Hide internals of various DetNet data plane from end-applications
  - End-systems are IP based, maybe design for IPv6 (no elegant way to extend on IPv4)
- Practical mapping of flow specific traffic treatment
  - In DetNet-IP [RFC 8939] section 5.3, flow id determines traffic treatment provisioned in the DetNet.
  - This could lead to scalability challenges [on going DP enhancements]
  - Limitations with service sub-layers
- Split Traffic flows
  - Architectural consideration. Most process automation is on-site,
  - With only support for remote monitoring
- Variety of traffic patterns within and for different {controller-field-device} pairs
  - Different latency bounds, urgent/alarm messages, closed control loops (bi-directional latency bounds) – these are per packet constraints
  - Generally long-lived DetNet flow reservations only provides coarse-granularity.

Generalization of interfaces
between the connected End systems

- Controllers
  - Associate with one of more field devices
  - All operations are controlled from these end systems
- Sensor
  - Emit operational data – periodically or event-based
  - Emit critical alarms.
- Actuating end-systems
  - Bring mechanical or physical changes to environment
  - Receive commands from controllers

- Common attributes
  - In general interface to actuators and sensors across different vendors and protocols is quite similar
  - Similar command- structure parameter changes
  - Represented as {controller, field-device} pair as communication endpoints.



Potential Traffic Patterns and Constraints

- Control Loops
  - To measure (sensor), compare (controller) and adjust (actuate) process variables
  - Each step is a separate instruction or packet as against a continuous flow.
- Periodicity
  - Many devices emit different type of readings with different interval
- Ordering
  - Must be preserved, out of order packets will be catastrophic to control loops
- Urgency
  - Failures and alarms must have highest precedence in the network.

- A deterministic network could support these patterns.
- Connect end systems to the network:
  - How end system can clearly communicate these without getting into the details of DetNet

Advantages of OCN Model

- Working with OCN Model allows automation applications to be designed closer to the real systems usually simplifying assumption
- Given such a model structure, traffic patterns and variables may be changed quickly
- May be able to provide better performance insights
- Validation can be achieved regardless of underlying technique used.

References

- OCN Github portal - Slide deck for the tutorial
  - https://github.com/kiranmak/ocn-tutorial-23
- Demo (work in Progress)
  - https://github.com/network2030/NewIP-Linux/tree/main
- DetNet/IETF Documents
  - RFC 9056 Deterministic Networking (DetNet) Data Plane: IP over MPLS
  - RFC 8964 Deterministic Networking (DetNet) Data Plane: MPLS
  - RFC 8939 Deterministic Networking (DetNet) Data Plane: IP
  - Using Deterministic Networks for Industry Operations and Control. Draft: https://datatracker.ietf.org/doc/draft-km-detnet-for-ocn-00
- Tintin - Tiny In-Network Transport for High Precision INdustrial Communication
  - https://ieeexplore.ieee.org/document/9940343
- Asymmetric addresses: Asymmetric Addressing Structures in Limited Domain Networks
  - https://ieeexplore.ieee.org/document/9481811
- Latency based forwarding Programmable Data Plane for New IP using eXpress Data Path (XDP) in Linux
  - https://ieeexplore.ieee.org/document/9831409
- New Services and Capabilities for Network 2030: Description, Technical Gap and Performance Target Analysis
  - https://www.itu.int/en/ITU-T/focusgroups/net2030/Documents/Deliverable_NET2030.pdf