IPv6 Future in Industrial Automation

Filip Zembok
Principal Development Engineer
Rockwell Automation

Jakub Korbel
Networks Architect
Rockwell Automation

Arun Guru
Principal Engineer/Systems Architecture
Kinetix Division, Rockwell Automation

Presented at the ODVA
2026 Industry Conference & 24th Annual Meeting
October 21, 2026
Vilamoura, Portugal

Abstract

As industrial automation systems continue to scale and increasingly integrate with enterprise IT infrastructures (for example in data centers), IPv6 is becoming a critical technology for the future evolution of the Common Industrial Protocol (CIP). IPv6 enables new approaches to scalability, address management, and network interoperability that are difficult to achieve with IPv4-based solutions.
Since the last ODVA Industry Conference, significant progress has been made toward introducing IPv6 support into EtherNet/IP and the entire CIP ecosystem, including CIP Safety. Building on the roadmap outlined in “An IPv6 Roadmap for EtherNet/IP”, this paper provides an update on the current state of the work and highlights recent technical results. It discusses the challenges encountered when designing IPv6-related CIP Objects, including object modelling, address management, and interactions with existing CIP services, and explains how these challenges have been addressed.
It also presents the technical details of device provisioning at scale using IPv6 within EtherNet/IP networks. It discusses how IPv6, combined with existing CIP mechanisms and well-known IP technologies, can be applied to simplify system deployment and commissioning workflows for customers, system integrators, and device vendors.

The paper concludes with a preview of the next steps on the IPv6 roadmap for EtherNet/IP and outlines areas of ongoing work, providing insight into how IPv6 capabilities are expected to evolve within the CIP ecosystem.

Keywords

IPv6, EtherNet/IP, CIP Safety, Industrial Automation, Device Provisioning, Topology-Aware Commissioning, SLAAC, LLDP, Hierarchical Network Autoconfiguration, Industrial Networking, IT/OT Convergence

 ©2026 ODVA, Inc.

Classified as Business

Definition of terms

Term	Definition
A (DNS)	DNS Address Record for IPv4
AAAA (DNS)	DNS Address Record for IPv6 (4x wider)
ACD	Address Conflict Detection. Mechanism used in IPv4 to detect duplicate IP addresses on a network.
APIPA	Automatic Private IP Addressing. Defined in RFC 3927.
CIP	Common Industrial Protocol.
CPF	Common Packet Format. Structure used in EtherNet/IP encapsulation messages to transport data items.
DAD	Duplicate Address Detection. IPv6 mechanism to verify uniqueness of an address as defined in RFC 4862.
DDNS	Dynamic Domain Name System. Mechanism to automatically update DNS records, defined in RFC 2136.
DHCPv6	Dynamic Host Configuration Protocol for IPv6.
DHCPv6-PD	DHCPv6 Prefix Delegation. Mechanism for delegating IPv6 prefixes to downstream networks.
DNS	Domain Name System. Distributed system for name resolution in IP networks.
Dual-Stack device	Device that supports both IPv4 and IPv6 simultaneously.
FOSS	Free and Open-Source Software
FQDN	Fully Qualified Domain Name. Complete domain name specifying a host within DNS hierarchy.
GUA	Global Unicast Address. Public IPv6 address routable on the Internet.
IA_NA	Identity Association for Non-temporary Addresses, i.e. DHCPv6-supplied IPv6 address.
IA_PD	Identity Association for Prefix Delegation, i.e. DHCPv6-supplied prefix pool.
IPv4	Internet Protocol Version 4.
IPv6	Internet Protocol Version 6.
IPv4-only device	Device that supports IPv4 and does not support IPv6.
IPv6-only device	Device that supports IPv6 and does not support IPv4.
IT	Information Technology, ie. hardware, software and communication technologies for general information processing
LLA	Link-Local Address. IPv6 address from fe80::/10 range, valid only on a local link.
LLDP	Link Layer Discovery Protocol. IEEE 802.1AB protocol used for discovering adjacent network devices.
mDNS	Multicast DNS. Link-local name resolution protocol defined in RFC 6762.
OT	Operational Technology, ie. hardware, software and communication technologies that detect or cause a change through the direct monitoring or control of industrial equipment.
PRP	Parallel Redundancy Protocol.
PTR (DNS)	DNS Pointer Record.

Classified as Business

QoS	Quality of Service.
RA	Router Advertisement. ICMPv6 message used to provide network configuration information to hosts.
RSTP	Rapid Spanning Tree Protocol.
SIG	ODVA Special Interest Group.
SLAAC	Stateless Address Autoconfiguration.
SNN	Safety Network Number.
SNCT	Safety Network Configuration Tool.
SNMP	Simple Network Management Protocol.
TRB	ODVA Technical Review Board.
ULA	Unique Local Address. IPv6 private address space defined in RFC 4193 (fc00::/7).
UNID	Unique Network Identifier. Identifier used in CIP Safety to uniquely identify a node in a safety system.

Introduction

The case for IPv6 keeps growing. Datacenters increasingly run IPv6-only fabrics because IPv4 address exhaustion makes large flat networks impractical, and the cloud platforms industrial systems connect to expect it. Legislators are pushing the same direction: government mandates in the US and elsewhere now require IPv6 on networked equipment [1], so vendors who skip it lose access to those markets. At the same time IT and OT are converging, and the IT side is already on IPv6. If industrial protocols stay IPv4-only they become the exception that needs special handling, which is exactly the kind of friction that IT/OT convergence is meant to remove. EtherNet/IP needs IPv6 to stay a first-class citizen on the networks it will run on, and because it is built on standard technologies, it is well-equipped to do so.

This paper continues the work started in "An IPv6 Roadmap for EtherNet/IP" paper presented at the 2025 ODVA Industry Conference [2] which was built on top of “EtherNet/IP over IPv6 - Evolution, not Revolution for the World's Leading Industrial Ethernet Variant” paper presented at the 2012 ODVA Industry Conference [3].

The roadmap laid out a phased plan for bringing IPv6 into the CIP Networks Library, from first proof of concept up to full functional parity with IPv4. This paper reports on where the work stands, what was done differently than planned, why those changes were made, and what has been validated already.

Classified as Business

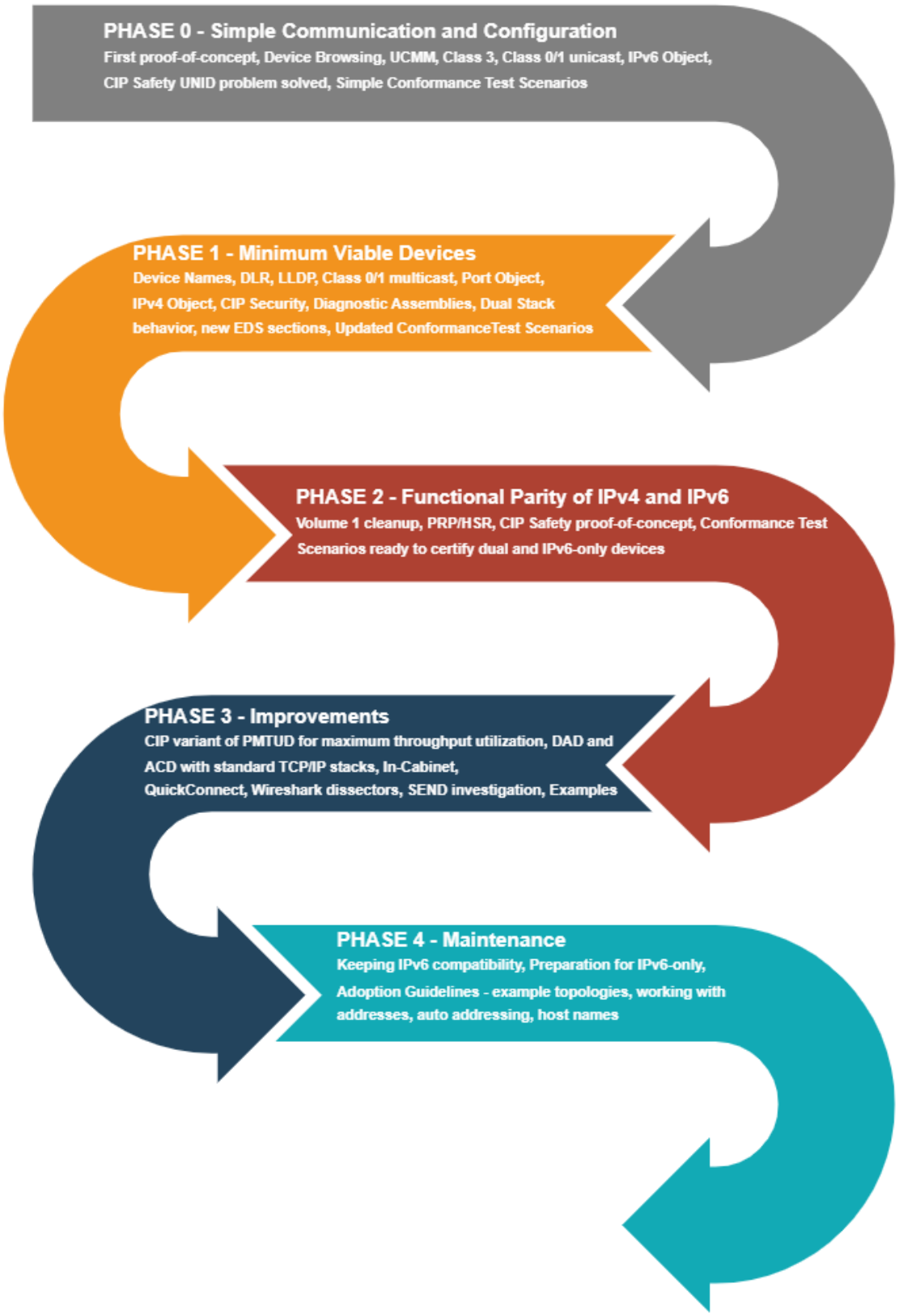


Figure 1 IPv6 Roadmap Phases

Classified as Business

We are still in Phase 0 according to the proposed roadmap shown in Figure 1. Specification work across several SIGs needs consensus, and the prototypes had to be built in parallel to keep the proposals honest.

This paper presents a concept-selection analysis for adding IPv6 support to CIP Safety and proposes a migration direction that preserves the existing UNID size while decoupling safety identity from network-layer addressing.

Additionally, this paper discusses advanced use cases involving IPv6, like topology-aware commissioning and possibility of building self-configuring hierarchies of industrial devices.

Phase 0 status

Device browsing is well defined (ESE-0001-089). The discovery flow works on both protocol families, and a tool can find IPv4-only, IPv6-only and dual-stack devices on the same link. The QoS object was generalized (ESE-0001-101) so it is no longer tied to IPv4 only, which was a minor edit, not the large rework that was anticipated.

For discovery the existing identity item was not overloaded. New List Identity items for IPv6 information decoupling address information from CIP Identity were added instead.

The biggest deviation from the roadmap is the configuration object. The roadmap proposed a single IPv6 object based on a standard YANG model. Instead, several new objects which follow the TCP/IP Interface object were designed trying to generalize its concepts, making them safer with atomic operations. An additional network interface object was added to anchor address objects. This keeps the model close to what implementers already know.

One planned item is still open. The link path was not updated for binary addresses, so CIP Routing into IPv6 networks is not possible yet, there is only a draft of a SE.

What was demonstrated

To verify validity of the proposed specification enhancements, main effort went into prototypes. At Automation Fair 2025 in Chicago and later to TRB we showed EtherNet/IP devices working over IPv6. The demo covered address autoconfiguration, host name assignment, and CIP Security configuration without manual entry. This confirms the concepts are realizable and gives the SIGs concrete behavior to specify against, which is exactly what Phase 0 was meant to produce.

Phase 0 progress summary

Table 1 Phase 0 Progress Summary

Capability	Current Status
CIP System Architecture SIG	
Usage of document-based configuration	Not planned
Standardization of Atomic Configuration changes	New objects use it, discussed on August 2026 Roundtable, plausible
Port Object Changes	Not started, planned
EtherNet/IP System Architecture SIG	
IPv6 Object	New objects proposals submitted to SIG (TBD: ESE)
Device Discovery	ESE-0001-089, SIG Approved

Classified as Business

Link Path update for binary addresses	Internal SE draft exists
CIP UCMM	Tested with multiple vendors (Rockwell, HMS) and multiple prototypes
CIP Class 0/1	Tested with multiple vendors (Rockwell, HMS) and multiple prototypes
CIP Class 3 connections	Tested with multiple vendors (Rockwell, HMS) and multiple prototypes
Specification Section for RFC 6434	It is obsoleted by RFC 8504, section planned
CIP Safety SIG	
Safety UNID problem	Solved, described in this paper. SIG subgroup is being created to update Volume 5
CIP Motion SIG	
Transparent clock implementation check	Not started
QoS Clarification	ESE-0001-101, SIG Approved

The sections that follow describe these results in more detail.

Specification enhancements

IPv4 vs IPv6 EtherNet/IP

The IPv6 enhancements to the CIP Specification should consider the following long-term factors:

1. The existing EtherNet/IP ecosystem is based on IPv4, and IPv4-only devices will continue to be present.
2. Some EtherNet/IP devices will operate in dual-stack mode, supporting both IPv4 and IPv6.
3. In distant future some EtherNet/IP devices will be designed as IPv6-only.

Backward compatibility is the primary motivation for the IPv6 enhancements described in subsequent sections. Existing IPv4 functionality is preserved without modification, while IPv6 capabilities are introduced in a way that leverages and extends established IPv4 constructs. Support for IPv6 is optional, allowing vendors to decide whether to incorporate it into their products based on their requirements and market needs.

Device discovery

EtherNet/IP provides a mechanism for discovering IPv4 devices on a local network segment. To identify all available IPv4 devices, a client issues a ListIdentity request Encapsulation Protocol command over IPv4 UDP broadcast. EtherNet/IP devices on the network respond directly to the originator of the request. The ListIdentity response includes a CIP Identity Item containing key device information, such as:

1. IPv4 socket address (IPv4 address and TCP port used for EtherNet/IP communication)
2. Encapsulation Protocol Version
3. Device Vendor ID
4. Device Type
5. Product Code
6. Device Status
7. Device Serial Number
8. Device Product Name
9. Device State

Most of the fields in CIP Identity Item correspond directly to attributes in the CIP Identity Object. The socket address is inherently tied to IPv4 and cannot be reused in an IPv6-based solution.

Classified as Business

In addition to the CIP Identity Item, the ListIdentity response may conditionally include other Common Packet Format items. These items may provide details such as supported EtherNet/IP Transports, EtherNet/IP Usage and CIP Security application profiles. The presence of these additional items is conditional and depends on the capabilities of the responding device.

The proposed mechanism for discovering IPv6 EtherNet/IP devices closely aligns with the existing IPv4 approach. To identify available IPv6 devices, a client issues the same ListIdentity request command but in case of IPv6 it sends it as a UDP datagram to the IPv6 link-local all-nodes multicast address (ff02::1). IPv6 EtherNet/IP devices respond directly to the originator using their IPv6 link-local address.

The ListIdentity response over IPv6 contains a new CPF item, the CIP Identity Core Item. This item is conceptually similar to the existing CIP Identity Item but excludes IPv4-specific elements and encapsulation protocol version information. Table 2 provides a comparison between the CIP Identity Item and the CIP Identity Core Item.

Table 2 Comparison of CIP Identity Item and CIP Identity Core Item

Member name	CIP Identity Item	CIP Identity Core Item
Item ID	Present	Present
Item Length	Present	Present
Encapsulation Protocol Version	Present	Removed. Not used.
IPv4 Socket Address	Present	Removed. Functionality migrated to IPv6 Address Item.
Vendor ID	Present	Present
Device Type	Present	Present
Product Code	Present	Present
Revision	Present	Present
Status	Present	Present
Serial Number	Present	Present
Product Name	Present	Present
State	Present	Present

To retain the functionality previously provided by the IPv4 socket address in the CIP Identity Item, a new CPF item is defined: the IPv6 Address Item. This item includes the IPv6 address and TCP port used for EtherNet/IP communication. The introduction of the IPv6 Address Item serves two key purposes:

1. Decouples address information from identity information.
2. Enables devices to advertise multiple IPv6 addresses, reflecting the fact that IPv6 devices can naturally have more than one address assigned to a single network interface.

The address information conveyed in both the IPv4 socket address and the IPv6 Address Item reflects the address information from the perspective of the device sending the response. The address information presented by the device may be different from the address visible form the client perspective in scenarios where Network Address Translation is configured between the client and the device.

Table 3 outlines which CPF items are permitted in ListIdentity responses for both IPv4 and IPv6.

Table 3 CPF items in IPv4 and IPv6 ListIdentity responses

CPF Item	IPv4	IPv6
CIP Identity Item	Required. Must be the first returned item.	Not allowed.

CIP Identity Core Item	Not allowed.	Required. Must be the first returned item.
EtherNet/IP Capability Item	Required if the device implements UDP-Only EtherNet/IP transport profile, optional otherwise.	
EtherNet/IP Usage Item	Required if the device implements In-Cabinet Usage profile, optional otherwise.	
CIP Security Information Item	Required if the device implements CIP Security, not allowed otherwise.	
IPv6 Address Item	Not allowed	There shall be one instance of this item provided for each IPv6 address used for EtherNet/IP on the network interface on which the ListIdentity request was received.

Overall, the proposed IPv6 EtherNet/IP device discovery mechanism maximizes reuse of the existing IPv4 solution. The ListIdentity request and response formats remain unchanged. The limits on simultaneous requests, as well as the behavior governing random response delays, are consistent across both IPv4 and IPv6. The primary differences between IPv4 and IPv6 lie in the specific CPF items that are mandatory or permitted within the ListIdentity response.

EtherNet/IP clients that support both IPv4 and IPv6 and intend to discover all EtherNet/IP devices on a local network segment shall issue a ListIdentity request over both protocols and consolidate the information received in the respective responses.
The ListIdentity requests over IPv4 (UDP broadcast) and IPv6 (link-local all-nodes multicast) may be sent in parallel. Responses can then be correlated and consolidated based on matching attributes - such as vendor ID, device type, and device serial number - obtained from the CIP Identity Item and CIP Identity Core Item.

IPv4 in EtherNet/IP

The IPv4 aspect of an IPv4-based EtherNet/IP solution is modelled by the TCP/IP Interface Object. Since the first EtherNet/IP specification was published in 2001, this object has been extended and improved few times. The most recent revision is version 4.

The TCP/IP Interface Object provides a comprehensive set of capabilities for the configuration, monitoring, and management of an IPv4-based EtherNet/IP interface. Its key functionalities include:

- TCP/IP network interface configuration (IPv4 address, network mask, default gateway)
- DNS configuration
- IPv4 multicast configuration (address allocation and TTL for multicast packets)
- IPv4 Address Conflict Detection diagnostics
- Firewall settings (IANA Port Administration, IANA Protocol Administration)
- EtherNet/IP settings (Encapsulation Inactivity Timeout)
- Diagnostics (active TCP connections, non-CIP encapsulation messages per second)
- Reporting association to the Ethernet Link Object

However, the TCP/IP Interface Object has several limitations:

- It supports association with only a single instance of the Ethernet Link Object, which is insufficient for DLR and PRP interfaces.
- It requires a specific order when setting the values of Attribute 3 (Configuration Method) and Attribute 5 (Interface Configuration) via Set_Attribute_Single service.
- The Host Name attribute does not function as a host name in terms of DNS and is limited to informational use.
- For devices that require a reset to apply new interface configuration, the object only signals that a reset is necessary; the pending configuration cannot be read back from the device.

Classified as Business

- Address Conflict Detection is only an optional feature. Although recommended, its absence increases the risk of network instability in cases of misconfiguration.

TCP/IP Interface Object is built around the single IPv4 address. Such approach is good enough for simple devices that offer one EtherNet/IP CIP communication port. Such approach is flexible however it raises some challenges for more complicated devices that can have multiple EtherNet/IP CIP communication ports and that can have multiple IPv4 addresses that share the same physical interface to the network.

IPv6 Link Local Interface and IPv6 Global Unicast Interface Objects

The proposed IPv6 CIP specification enhancements introduce two new objects: the IPv6 Link Local Interface Object and the IPv6 Global Unicast Interface Object.

Both objects are conceptually aligned with the existing TCP/IP Interface Object. They enable configuration and diagnostics of EtherNet/IP behavior on a device's TCP/IP network interface. Due to inherent differences in the feature sets of IPv6 link-local and global unicast address types, separate objects have been defined for each address type.

The design of both IPv6 objects follows these key principles:

1. **Maintain the TCP/IP Interface Object model of one object instance per one IP address.**

   Alternative approaches for modelling IPv6 functionality within the CIP specification were evaluated, including:
   - One IPv6 Object instance for managing all IPv6 addresses on a network interface.
   - One IPv6 Object instance with paired link-local and global unicast address. Such a pair of addresses is functionally close to one IPv4 address.

   Ultimately, the one object instance per one IPv6 address approach was selected, as it provides the greatest flexibility for vendors and supports a more modular and scalable implementation model.

2. **Leverage existing definitions. Where applicable reuse attributes and services from the TCP/IP Interface Object.**

   Following attributes of TCP/IP Interface Object were identified as one-to-one reusable by new IPv6 objects:
   - Encapsulation Inactivity Timeout
   - IANA Port Admin
   - IANA Protocol Admin
   - Active TCP Connections
   - Non-CIP Encapsulation Messages Per Second

   Following services of TCP/IP Interface Object were identified as one-to-one reusable by new IPv6 objects:
   - Set_Port_Admin_State
   - Set_Protocol_Admin_State

   Those services control EtherNet/IP device behavior that is IP protocol version independent.

   Following other elements of TCP/IP Interface Object were identified as reusable by new IPv6 objects:
   - Diagnostics Connection Point definition. This Connection Point is composed of attributes of TCP/IP Interface Object that are identified as one-to-one reusable by new IPv6 Objects; thus, the entire Connection Point can be reused.
   - CIP Security clarifications. This section of TCP/IP Interface Object discusses recommended factory-default UDP and TCP port settings for EtherNet/IP devices. Those recommendations are IP protocol version independent therefore apply equally to EtherNet/IP over IPv6.

Classified as Business

- Error codes. The error codes defined in TCP/IP Interface Object are applicable to Set_Port_Admin_State and Set_Protocol_Admin_State services, both of which have been identified as one-to-one reusable by new IPv6 Objects.

The common, reusable elements have been extracted from the TCP/IP Interface Object and consolidated into a dedicated chapter within the specification. The TCP/IP Interface Object, IPv6 Link-Local Interface Object, and IPv6 Global Unicast Interface Object only declare support for the relevant attributes, services and connection points and reference their common definitions in that chapter.

3. **Avoid creating duplicate definitions. Introduce new IPv6 definitions in a manner that enables reuse across new objects.**

   All newly introduced reusable elements are defined once in a dedicated chapter within the specification. These elements include:
   - IPv6 address status enumeration
   - Object state machine
   - Object-specific services governing transitions between state machine states

4. **Address known limitations of TCP/IP Interface Object. Avoid the repetition of existing design shortcomings and introduce improvements that resolve issues listed in IPv4 in EtherNet/IP chapter.**

   New IPv6 objects share the state machine that is shown in Figure 2.

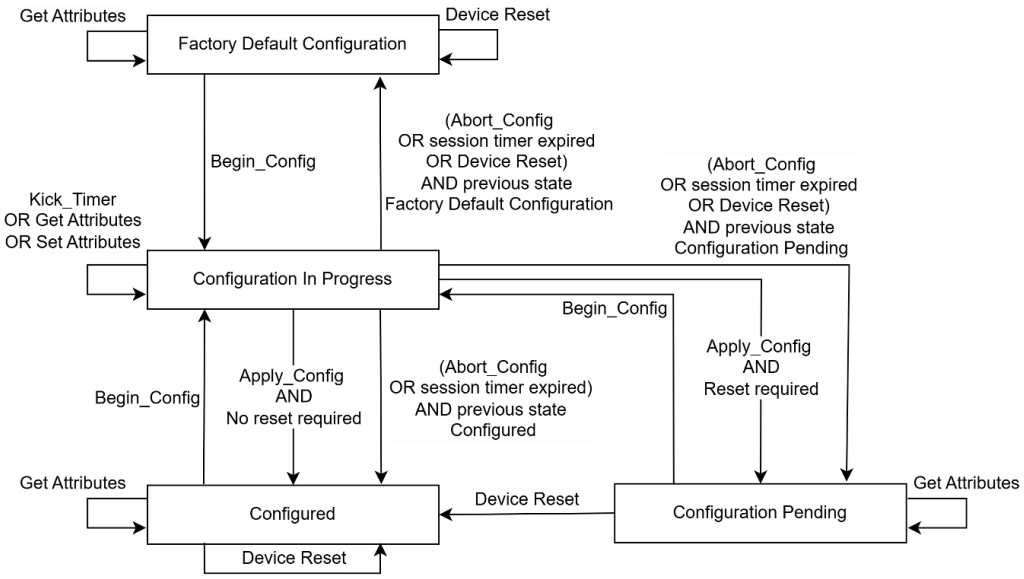


Figure 2 State machine used by new IPv6 objects

The presented state machine is inspired by similar designs used in CIP Security objects. It controls when object modifications are permitted and how they are applied.

To initiate any changes to an object instance, the Begin_Config service must be invoked. This service starts a configuration session and transitions the object into the Configuration in Progress state. It also activates a timer that limits the duration of this session. Once configuration session is started the new instance settings can be prepared. The object accepts modification services; however, the changes are not applied immediately. Instead, they are stored as pending values.

Classified as Business

The configuration session can be finished in several ways:

- Applying the pending configuration using the Apply_Config service.
- Aborting the session with the Abort_Config service - all pending settings are discarded.
- Allowing the session timer to expire - all pending settings are discarded.

When Apply_Config service is called, two outcomes are possible:

- If no device reset is required for the new settings to take effect, the object instance is immediately reconfigured and transitions to the Configured state.
- If the object instance cannot be reconfigured immediately, the object transitions to the Configuration Pending state. This state informs the user that a device reset is required for the new configuration to be fully applied.

Presented state machine mitigates the shortcomings of approach that require certain attributes of the TCP/IP Interface Object to be set in a predefined order.. The order in which new settings are delivered becomes irrelevant, as all changes are applied together upon Apply_Config.

The new IPv6 objects decouple current object instance settings from pending settings. The object state machine governs when active and pending settings are available. When an object is queried while in the Configuration in Progress state, it returns the pending values, otherwise it returns values currently applied on the device.

This approach removes ambiguity by clearly distinguishing between the device's active configuration and any changes awaiting application.

Address collisions

The IPv6 counterpart of IPv4 Address Conflict Detection (ACD) is Duplicate Address Detection (DAD). In IPv4, ACD was specifically adapted for CIP EtherNet/IP, which introduced several implications for vendors. Notably, standard ACD algorithms provided by operating systems could not be used directly, requiring each vendor to develop a custom implementation tailored to EtherNet/IP requirements.
For IPv6, the preferred approach is to adopt the Duplicate Address Detection procedure defined in RFC 4862 without modification, as it has been evaluated as sufficiently for use in EtherNet/IP environments. Furthermore, DAD is made required and by default enabled for EtherNet/IP over IPv6, reducing the risk of network instability in cases of misconfiguration. To support diagnostics, the IPv6 Link Local Interface Object and IPv6 Global Unicast Interface Object provide DAD related information comparable to the diagnostics offered by the TCP/IP Interface Object for IPv4 ACD.

EtherNet/IP Interface Object

The coexistence of IPv4 and IPv6 is a new concept within EtherNet/IP. Introducing IPv6 adds complexity, including the need to manage multiple IPv4 and IPv6 addresses simultaneously. IPv6 also introduces link-local addresses with their distinct characteristics. IPv6 link-local addresses have link scope, meaning they are valid only on a single network interface or link. As a result, they require an associated identifier (anchor) to define and disambiguate their scope within the system.

The current EtherNet/IP specification does not define a "network interface" concept - an entity that serves as a common anchor for multiple TCP/IP interfaces and enables unified view and management of their shared EtherNet/IP settings.
The existing CIP objects do not adequately fulfil this role.
The Port Object is conceptually positioned above the TCP/IP Interface. However, this object describes the device at the application layer and abstracts from specific network technology details. As a result, it is not suitable for configuring EtherNet/IP-specific settings, like for example enabling or disabling IPv4 and IPv6.

Classified as Business

The Ethernet Link Object to some degree defines a network interface in term of the link scope. However, this object conceptually operates below the TCP/IP Interface and is dedicated to OSI Layer 2 functionality. Consequently, it is also not appropriate for managing EtherNet/IP settings at the interface level.

EtherNet/IP devices are evolving to become more complex, with an increasing number of TCP/IP interfaces. It is expected that future devices will include multiple physical Ethernet ports and a growing number of TCP/IP interfaces both IPv4 and IPv6 based, further increasing configuration and management complexity.

While these challenges are new to the EtherNet/IP ecosystem, they have already been addressed in other domains. Therefore, the EtherNet/IP approach can draw on established practices from other industries.

The IPv6 CIP Specification enhancement proposes the EtherNet/IP Interface Object. This new object is conceptually aligned with:

- the “ifconfig” command in Linux,
- the “ipconfig” command in Windows, and
- the “Network Connections” interface in the Windows Control Panel.

The tools enumerated above are centered on the concept of a network interface and enable users to view, diagnose, and manage IP configuration and other network features across multiple interfaces. This includes the ability to enable or disable IPv4 and IPv6 on a per-interface basis, as well as to configure, manage and diagnose additional network features for each interface.

The EtherNet/IP Interface Object aims to provide a subset of the capabilities of the tools enumerated above within the EtherNet/IP framework.

The EtherNet/IP Interface Object instances offer following capabilities:

- Enable or disable IPv4 and IPv6 protocols.
- Enable or disable IPv4 and IPv6 network interface level features.
- Provide network interface name consistent with the physical labelling on the device.
- Provide association with Ethernet Link Object instances. The association is flexible, it supports known two Ethernet ports configurations like DLR and PRP, while remaining extensible to accommodate more complex configurations in the future. This resolves a known limitation of the TCP/IP Interface Object described in earlier sections.
- Provide association with instances of objects that define and describe EtherNet/IP functionalities on the network interface, such as the DLR Object and LLDP Data Table Object.
- Act as an anchoring point for EtherNet/IP TCP/IP Interfaces (TCP/IP Interface Object, IPv6 Link Local Interface Object and the IPv6 Global Unicast Interface Object) that operate on the network interface.
- Provide a high-level diagnostic indicator reflecting the status of the network interface. This status can serve as a trigger to obtain comprehensive network interface diagnostics from associated EtherNet/IP objects.

The EtherNet/IP Interface Object instances are governed by the same state machine as IPv6 Link Local Interface and IPv6 Global Unicast Interface objects.

The concept of the EtherNet/IP Interface Object is most effectively illustrated through a diagram. Figure 3 presents the relationships between CIP objects of an example device that provides various communication capabilities via EtherNet/IP.

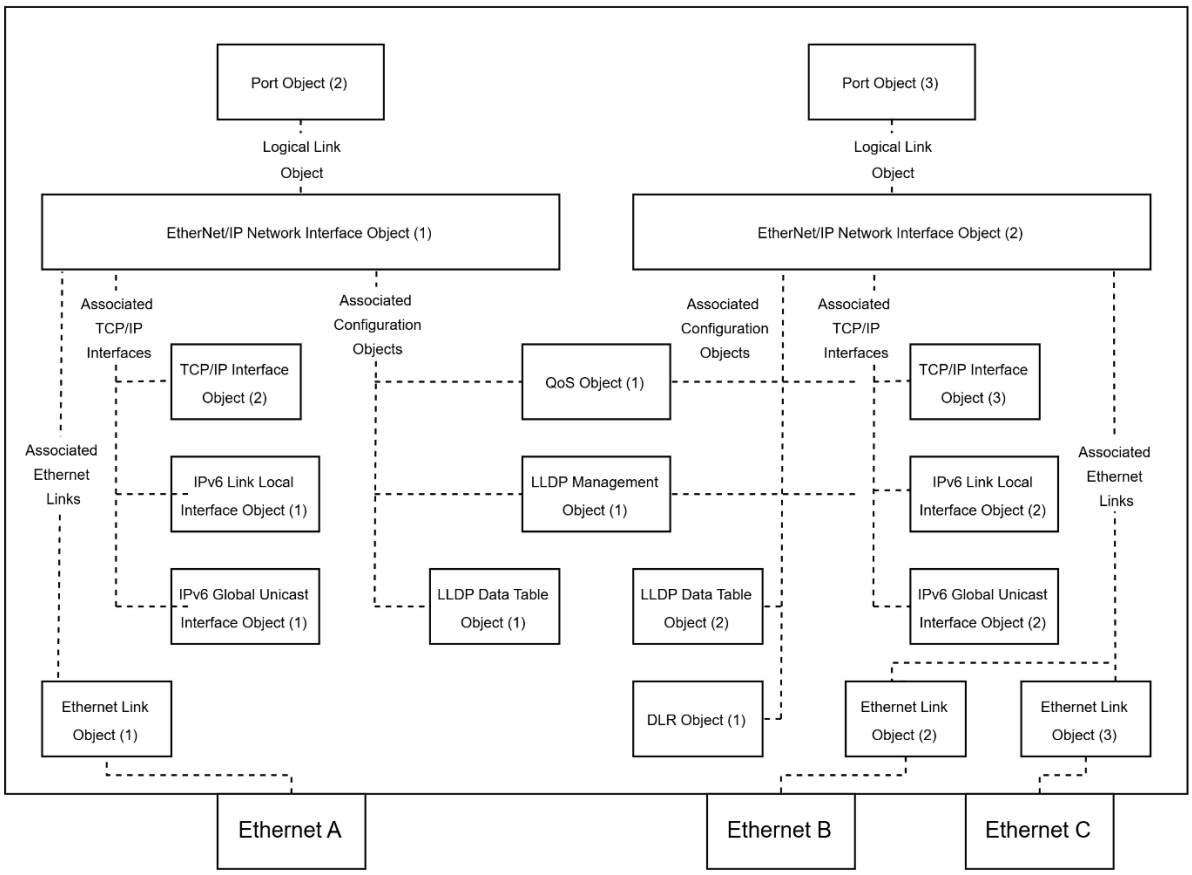


Figure 3 Example device with various EtherNet/IP communication capabilities

The example device in Figure 3 is equipped with three Ethernet ports and two EtherNet/IP Network Interfaces.

The first EtherNet/IP Network Interface of the example device is described by EtherNet/IP Network Interface Object instance 1. This interface operates on a single Ethernet port. This instance is identified by the name “A,” which is derived from the Interface Label attribute of the associated Ethernet Link Object.

There are three IP addresses anchored to EtherNet/IP Network Interface Object instance 1:

- one IPv4 address, governed by TCP/IP Interface Object instance 2.
- one IPv6 link-local address, governed by IPv6 Link Local Object instance 1.
- one IPv6 global unicast address, governed by IPv6 Global Unicast Interface Object instance 1.

Additionally, EtherNet/IP Network Interface Object instance 1 provides associations to instances of objects used for configuring and diagnosing EtherNet/IP behavior on this network interface. These include:

- QoS Object instance 1.
- LLDP Management Object instance 1.
- LLDP Data Table Object instance 1.

The second EtherNet/IP Network Interface of the example device is represented by EtherNet/IP Network Interface Object instance 2. This interface uses two Ethernet ports. The instance is named “B/C,” its name is a combination of the Interface Label attributes of the associated Ethernet Link Object instances.

There are three IP addresses anchored to EtherNet/IP Network Interface Object instance 2:

- one IPv4 address, governed by TCP/IP Interface Object instance 3.
- one IPv6 link-local address, governed by IPv6 Link Local Object instance 2.
- one IPv6 global unicast address, governed by IPv6 Global Unicast Interface Object instance 2.

The EtherNet/IP Network Interface described by EtherNet/IP Network Interface Object instance 2 utilizes two Ethernet ports to provide DLR functionality. DLR Object instance 1 is enumerated alongside QoS Object instance 1, LLDP Management Object instance 1, and LLDP Data Table Object instance 1 within the associated configuration objects.

It should be noted that QoS Object instance 1 and LLDP Management Object instance 1 are shared across both network interfaces, as the settings of these object instances apply to the entire device.

The one-to-one relationship between a Port Object instance and an EtherNet/IP Network Interface Object instance is intended to simplify the use of EtherNet/IP. The EtherNet/IP Network Interface Object instance can be the value of the Logical Link Object attribute of the Port Object.

With this configuration, Port Segments in CIP route paths can use the same Port Identifiers regardless of the IP address type used as the Link Address. As a result, CIP communication can be directed to the EtherNet/IP network connected to Ethernet port A of the device shown in Figure 3 using the following addressing formats:

- 2, 192.168.1.10
- 2, fe80::20
- 2, 2001:1234:5678::20

The source IP address for such communication can be determined based on the target address.

For devices that support multiple IPv4, IPv6 link-local, or IPv6 global unicast addresses on a single network interface, stricter control of the source address for outgoing CIP communication is possible. In such cases, this can be achieved by defining multiple Port Object instances and associating them with specific instances of the TCP/IP Interface Object, IPv6 Link Local Object, or IPv6 Global Unicast Interface Object.

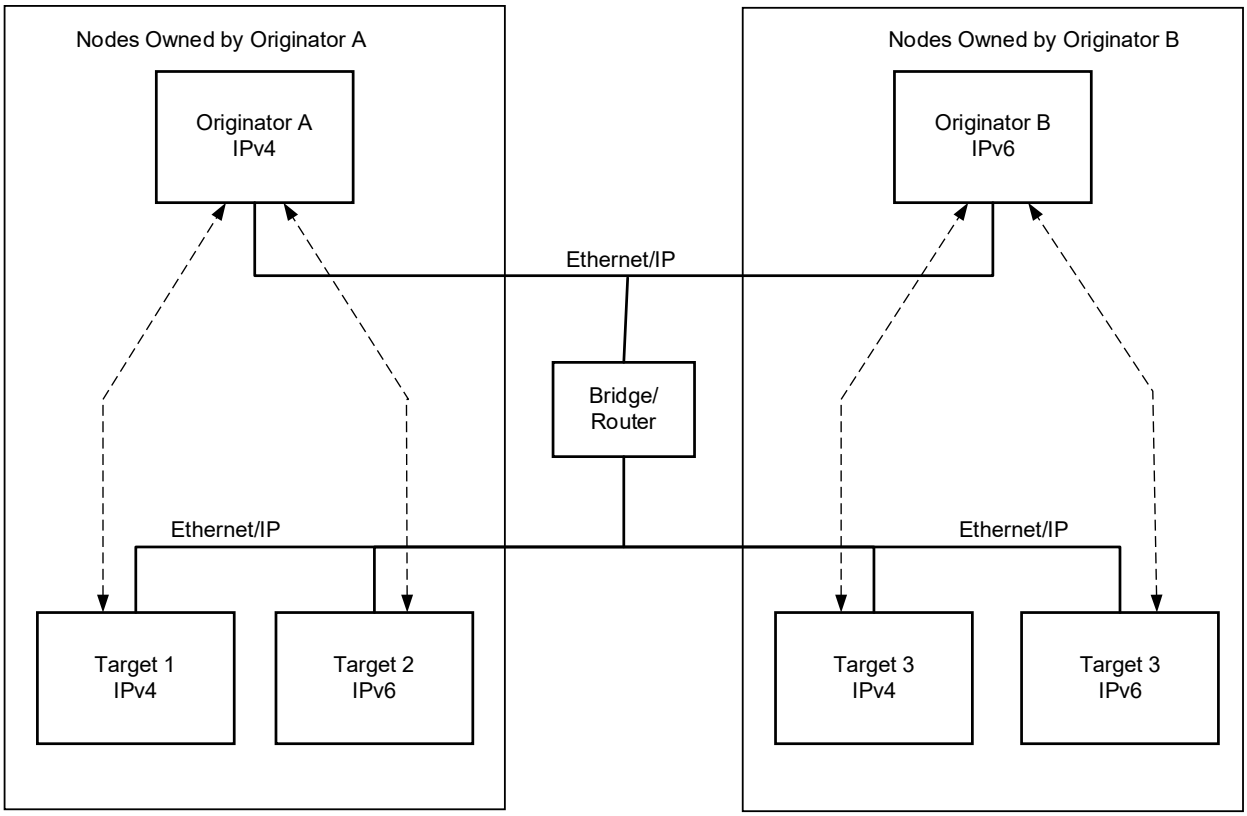
IPv6 vs Safety

CIP Safety was designed to provide safety communication over CIP networks using a safety-layer architecture that is logically independent of the underlying communication channel. A central element of this architecture is the System-Wide Unique Node Identifier (UNID), which uniquely identifies a node across all networks in the safety system. In the current model, the UNID consists of a 6-byte Safety Network Number and a 4-byte NodeID with the NodeID sized to accommodate existing CIP network address forms. For EtherNet/IP safety devices, the NodeID has historically aligned well with IPv4 addressing because an IPv4 address is four bytes and uniqueness can be ensured by mechanisms such as Address Conflict Detection (ACD).

The transition to IPv6 introduces a mismatch between the existing UNID structure and the size and semantics of IPv6 addresses. IPv6 addresses are 16 bytes and are commonly structured as a 64-bit subnet prefix and a 64-bit interface identifier. The interface identifier alone does not guarantee system-wide uniqueness because uniqueness depends on the subnet prefix as well as the host identifier. Therefore, simply placing part of an IPv6 address into the existing 4-byte NodeID field is insufficient unless additional uniqueness rules, services, or validation mechanisms are introduced.

The challenge is not merely one of field size. CIP Safety must preserve interoperable safety behavior across legacy IPv4 devices, new IPv6-capable devices, and mixed systems that may include originators,

targets, multi-port devices, software tools, and devices from multiple vendors as shown in Figure 3 below. The selected approach must also avoid imposing excessive configuration burden on users and should minimize changes to safety specifications, firmware, software tools, and certification artifacts.



(Dotted lines represent logical relationships/
connections, solid lines are physical connections)

Figure 4: Safety system with originating and target devices provisioned with IPv4 and IPv6 addresses.

The core problem is that the current CIP Safety UNID allocates only four bytes for the host identifier, while IPv6 requires substantially more address space to represent an address or address-derived identity that is unique globally. The CIP Safety SIG identified three broad design choices:

1. Extending the Safety Open network segment to support full IPv6 address information.
2. Avoid extending the segment by inventing a unique 4-byte or 8-byte value.
3. Extending the segment while using another unique method that is not directly address-related and still supports device replacement.

The design must balance competing requirements. Expanding the UNID to include larger IPv6-derived fields is straightforward conceptually, but it creates backward-compatibility and specification-change implications. Conversely, preserving the existing 10-byte UNID size improves backward compatibility but requires a defensible (to certifying bodies such as TUV) uniqueness strategy for IPv6-capable devices. The selected concept must also work in multi-vendor environments, where tools, devices, and originators may be supplied by different vendors and where responsibility for uniqueness cannot be assumed to reside in a single proprietary workflow.

Classified as Business

Customer impact is especially important because the CIP Safety SIG assigned high relative weight to customer impact and UNID uniqueness in the selection criteria. A technically elegant solution that increases user configuration burden, creates difficult device-replacement workflows, or complicated mixed IPv4/IPv6 deployments may be less attractive than a solution that preserves familiar operational workflows while adding clean IPv6 support.

Concept 1: Expand the UNID to Accommodate IPv6

The baseline concept expands the UNID or Safety Open network segment to accommodate IPv6-derived information. This approach is straightforward because it directly represents larger IPv6 identity information rather than compressing or abstracting it.

The primary drawback is the impact on message size and specification compatibility. Increasing the safety segment size to accommodate larger target and originator UNIDs can reduce available safety-configuration assembly space and may affect legacy devices near size limits. Changing the NodeID portion of the UNID to support a full 16-byte IPv6 address would require changes to multiple CIP Safety messages, including SafetyOpen or type 1 Forward_Open with Safety Network Segment and SNCT messages.

CIP Safety SIG refined this concept by noting that the baseline concept could minimally expand the existing UNID structure from 10 bytes to 14 bytes to include the interface identifier portion of the IPv6 address along with the existing SNN fields. We also noted that added size in the Safety Network Segment may be offset by removing fields associated with time correction that are not used in the same way for the target use case, but it would still require a new Safety Network Segment type for IPv6.

Concept 2: Tool-Assigned Identifier Decoupled from IP Address

Another candidate retains the existing SNN structure while replacing the IPv4-address-derived NodeID with a tool-assigned value that is not coupled to the IP address. This approach allows the safety identity to be decoupled from the network-layer address. There would no longer be a need for the device to verify the NodeID against the physical IP address assigned to that device.

Concept 2 would use a software tool to generate a 4-byte random number to replace the existing IPv4 address fields of the UNID while retaining the 6-byte SNN type, date, and time fields to ensure uniqueness. A new SNN type would be needed so IPv6 devices do not perform the IPv4 address validation required today on IPv4 networks. The concept can reduce coupling to IP addressing and may be advantageous for systems where network addresses change, where device replacement is supported by tooling, or where future multi-port devices need a stable safety identity independent of port-level addressing.

The risk is that the approach relies on software tools to generate unique random numbers and may add burden to users when replacing failed devices in the field. It also raises questions about whether software tools generating random 4-byte addresses require safety certification, whether common ODVA-owned tools or multiple vendor-specific tools would be used, and how interoperability would be ensured.

Concept 3: Hash-Based 8-Byte Identifier

The hash concept compresses IPv6-related information into an 8-byte field. The proposed hash inputs include the interface identifier, date/time information, and SNN type, with candidate algorithms including MurmurHash3, CityHash, or a KDF. Hash collisions could be checked against known address and

datetime information but raises the question of whether an 8-byte hash with additional verification is sufficient to assure uniqueness

We further considered this concept in more detail and noted that hashing techniques are well known and that a common implementation could reduce vendor variability. Hash concept may compare favorably against a tool-generated random-number concept because implementation would be in safety firmware and vendors could adopt common code. Nevertheless, the concept still requires a convincing uniqueness argument and introduces the challenge of proving that the selected hash size and verification model are adequate for the safety identity use case.

Concept Selection

The concept-selection work done by the CIP Safety SIG identified weighted criteria for comparing proposed solutions as shown in Table 4. The criteria include firmware-update effort, solution complexity, specification effort, customer impact, backward compatibility, software-tool effort, uniqueness of the UNID, and future proofing. Customer impact and UNID uniqueness carry the highest relative weights among the listed criteria, making usability and safety-identity robustness dominant considerations in the final selection.

Table 4 Weighted Criteria for Comparing Proposed Solutions for adding IPv6 to CIP Safety

Key Criteria (Weighted Importance)	**Concept 1**	**Concept 2**	**Concept 3**
Customer Impact (HIGH)	❌ Higher disruption	⚠️ Moderate	⚠️ Moderate
UNID Uniqueness (HIGH)	✅ Strong deterministic	⚠️ Tool-dependent	⚠️ Statistical
Backward Compatibility (HIGH)	❌ Weak	✅ Strong	✅ Strong
Future Proofing	✅ Strong	✅ Strong	✅ Strong
Spec Effort	❌ High	✅ Low	⚠️ Medium
Firmware Effort	⚠️ Medium	✅ Low	⚠️ Medium
Tool Effort	✅ Low	❌ High	⚠️ Medium
System Complexity	⚠️ Moderate	✅ Lower (simplifies identity management)	⚠️ Moderate

IPv6 CIP Safety – Decision Matrix

The CIP Safety SIG analyzed concepts enumerated above in detail using a PUGH matrix and ranked them using the weighted criteria discussed in the previous section and Table 5 summarizes the three

concepts after they were scored. Table 6 summarizes relative strengths and risks associated with the three concepts in qualitative terms.

Table 5 Comparison of three concepts for adding IPv6 to CIP Safety

Criteria	Concept 1: Expanded UNID	Concept 2: Decoupled UNID (Tool Generated Random Number)	Concept 3: Hash-based UNID (IPv6-dervied)
Core Idea	Extend UNID to include IPv6 information	Remove IP dependency; assign random/tool-generated ID	Compress IPv6 into fixed-size hash
Alignment to IPv6	✅ Native (direct mapping)	✅ Abstracted (independent of IPv6)	⚠️ Indirect (derived via hashing)
Backward Compatibility	❌ Requires new formats and message changes	✅ Preserves 10-byte UNID structure	✅ Preserves size, minimal structural change
Uniqueness Guarantee	✅ Strong (full address)	⚠️ Depends on tool + ACD enforcement	⚠️ Depends on hash collision bounds
Customer Impact	⚠️ Medium–High (firmware + system changes)	⚠️ Medium (tooling/process changes)	⚠️ Medium (trust in algorithm)
Spec Impact	❌ High	✅ Low	⚠️ Medium
Firmware Complexity	⚠️ Medium	✅ Low	⚠️ Medium
Tooling Dependency	✅ Low	❌ High (UNID management + persistence)	✅ Low–Medium
Device Replacement	✅ Good (address-based)	✅ Good (if persisted by tool)	⚠️ Moderate (recompute/hash consistency)
Multi-Port Device Support	❌ Complex (multiple UNIDs per IP type)	✅ Strong (identity independent of ports/IPs)	✅ Moderate
Future Proofing	⚠️ Medium	✅ Strong	⚠️ Medium
SIG Direction	Baseline for comparison	**Top-ranked candidate**	Alternative contender

The evaluation criteria reflect the practical realities of standards evolution in a deployed industrial ecosystem. Firmware effort matters because both originators and targets may require updates to support

Classified as Business

new formats or services. Specification effort matters because changes to Safety Open, Safety Network Segment formats, SNCT messages, or related services can affect conformance and certification efforts. Software-tool effort matters because a consistent tool behavior may be required to generate, store, validate, or restore UNIDs, especially during commissioning and replacement. Future proofing matters because future devices may include multi-port Ethernet designs that support IPv4 and IPv6 on the same or different ports.

Table 6 Relative strengths and risks associated with each of the three concepts

Concept	Strengths	Risks / Open Issues	Overall Assessment
Concept 1 – Expanded UNID	✅ Direct IPv6 representation ✅ No ambiguity in address-to-identity mapping ✅ Deterministic uniqueness	⚠️ Breaks existing message formats ⚠️ Adds complexity for multi-port devices ⚠️ Complicates mixed IPv4/IPv6 networks ⚠️ Increases safety payload footprint	**Lower strategic fit** because it ties safety identity too closely to the network layer and creates backward-compatibility concerns
Concept 2 – Decoupled UNID	✅ Strong black-channel alignment: identity ≠ network ✅ Maintains existing 10-byte UNID ✅ Backward compatible ✅ Simplifies multi-port devices ✅ Supports dual-stack IPv4/IPv6 deployments ✅ Flexible for future network models	⚠️ Requires robust tooling ecosystem ⚠️ Needs UNID assignment and lifecycle management ⚠️ Persistence strategy must be defined ⚠️ Open question: who guarantees uniqueness across vendors?	**Best emerging option** because it preserves compatibility while aligning cleanly with black-channel safety principles
Concept 3 – Hash-Based UNID	✅ Deterministic: same input → same identity ✅ Avoids external tooling dependency ✅ Preserves existing 10-byte UNID ✅ Useful for automated provisioning models	⚠️ Collision risk must be formally justified ⚠️ Adds certification burden ⚠️ Proof of uniqueness bounds required ⚠️ Less intuitive for users and debugging	**Promising but certification-heavy** because uniqueness depends on proving acceptable collision probability

The recommended direction is to adopt an IP-address-decoupled UNID scheme for IPv6-capable CIP Safety devices while preserving legacy IPv4-based UNID behavior. This recommendation is grounded in the SIG discussion that Concept 2, which decouples the network address portion from the UNID, appeared more manageable than approaches requiring support for multiple IPv4 and IPv6 UNID formats

Classified as Business

on multi-port originators. It should also be noted that a baseline approach may require managing IPv4 UNIDs, IPv6 local UNIDs, and IPv6 global UNIDs, making a decoupled approach more manageable.

The preferred architecture should preserve three principles:

1. **Safety identity should be stable across network-address changes.**
   Decoupling the UNID from the IP address prevents changes in IPv6 addressing semantics, prefix delegation, link-local versus global addressing, or multi-port address assignment from forcing unnecessary changes in safety identity. This aligns with the note that decoupling CIP Safety from IP addresses allows the network to be treated more as a black channel.

2. **Backward compatibility should be preserved wherever possible.**
   The legacy SNN-plus-IPv4-address scheme should remain valid for existing IPv4 devices, while IPv6-capable devices support a new scheme or service. The legacy IPv4 scheme would remain and be allowed, and that everything else remains backward compatible, including Safety Open UNIDs remaining the same.

3. **Device replacement should be supported by tools or originators.**
   The software tool or originator can remember UNIDs for device-replacement scenarios. This is important because MAC-based and randomly generated identity schemes can create replacement challenges if the identity cannot be restored or reassigned predictably.

In conclusion, IPv6 support for CIP Safety requires more than increasing the size of the address field. The existing UNID model is tightly aligned with IPv4-era assumptions, where a 4-byte NodeID can naturally represent an IPv4 address. IPv6 introduces larger address structures, multiple address scopes, and interface identifier semantics that do not map cleanly into the existing 4-byte host identifier

Among the concepts reviewed, direct expansion of the UNID is straightforward but introduces larger message formats and broader specification impact. Interface-ID-only and MAC-address-based approaches are weakened by uniqueness and device-replacement concerns. Hash-based approaches offer a technically interesting compromise but require a strong uniqueness and collision-handling argument. The strongest direction is a backward-compatible, IP-address-decoupled UNID assignment model using a new proposal service for IPv6-capable devices while preserving the existing IPv4 scheme.

This direction aligns CIP Safety with the black-channel principle by separating safety identity from network-layer address representation. It also improves manageability for mixed IPv4/IPv6 and multi-port originator scenarios by avoiding the need to encode every address type directly into the safety identity. Future work should finalize the new proposal-service behavior, define duplicate-detection and replacement workflows, clarify SNN semantics in Volume 5, and establish interoperability requirements for software tools and originators.

Device provisioning with IPv6

IPv6, when combined with existing Common Industrial Protocol (CIP) mechanisms and well-established IP technologies, provides a foundation for simplifying system deployment and commissioning workflows. The device provisioning approach described below benefits customers, system integrators, and device vendors by reducing manual configuration effort and improving consistency across installations.

The discussed mechanism for device provisioning is based on two pillars:

1. Link Layer Discovery Protocol - LLDP

Classified as Business

2. IPv6 Stateless Address Autoconfiguration - SLAAC

The LLDP and SLAAC are described in following chapters.

LLDP

The LLDP is defined by IEEE 802.1AB. LLDP is an OSI Layer 2 protocol. It enables network devices to advertise identity, capabilities, and connectivity information to directly connected neighbors. This information typically includes device identification, port identification, system capabilities, and management addresses. As OSI Layer 2 protocol LLDP operates without requiring prior IP configuration, making it particularly suitable for early-stage network discovery.
CIP leverages the existing LLDP standard and extends it with ODVA-specific enhancements, as defined in [4]. LLDP was introduced into the CIP specification in November 2019 and has since been progressively made mandatory for an increasing range of device types.
In industrial automation systems, LLDP enables controllers, switches, and field devices to exchange topology information. This information can be accessed through the CIP LLDP Data Table Object.

Access to LLDP discovery results via the LLDP Data Table Object requires communication at OSI layers above Layer 2. Specifically, in an industrial EtherNet/IP network, this requires the device to have a configured IP address, as well as an initialized EtherNet/IP and CIP stack bound to that address.

SLAAC

In an IPv4-only environment, initial IP address configuration approaches vary across devices and may be complex and time consuming, particularly in large scale installations. The introduction of IPv6, with its built-in SLAAC capability, can significantly simplify this process.

SLAAC is a core IPv6 mechanism for address configuration and management. SLAAC can be considered an inherent feature of IPv6, as it is widely supported by IPv6-capable network stacks. One of its key functions is the ability for a device to independently and automatically determine the IPv6 link-local address for each of its interfaces. In addition, IPv6 SLAAC supports the creation of global unicast addresses based on Router Advertisement (RA) messages and ensures address uniqueness through the Duplicate Address Detection (DAD) procedure.

The IPv6 link-local address is formed by combining the prefix fe80::/64 with an interface Identifier. Devices may generate the interface Identifier using several methods, including:

1. Modified EUI-64 based on the device MAC address.
2. Stable, randomly generated identifiers [5].
3. Temporary or privacy-enhanced addresses [6].

The ability to automatically and independently determine the IPv6 link local address enables devices to become network-accessible immediately upon connection to the network. Once an IPv6 link-local address is assigned the operability of its EtherNet/IP and CIP stack is enabled. With that, the device can be discovered with IPv6 ListIdentity Encapsulation Protocol command and device’s LLDP Data Table Object can be queried for LLDP results.

LLDP + SLAAC practical application

The combination of LLDP and automatically generated IPv6 link-local addresses enables connected clients to discover both device presence and physical network topology without any prior configuration of the devices in the network. This capability allows for topology-aware provisioning workflows. The discovered topology information can be used as a reliable input for automation tools and commissioning systems.

Classified as Business

The practical use cases include but are not limited to:

1. Configuration deployment. Deploying the configuration from network design application to the machine being built.
2. Configuration replication. Capturing configuration from a reference machine and automatically deploying it to multiple identical systems.
3. Topology validation. Verifying the correctness of physical network connections by comparing the discovered topology against a predefined reference model.
4. Automatic configuration of a replacement device. Automatically configuring a device that is inserted into the network in place of a failed unit, based on its detected position within the topology.

All the practical use cases are built on following mechanism:

The prerequisite is that the machine network configuration is defined. The machine network configuration can be designed in network design applications or captured from the reference machine. The network configuration for the example machine is shown in Figure 5.

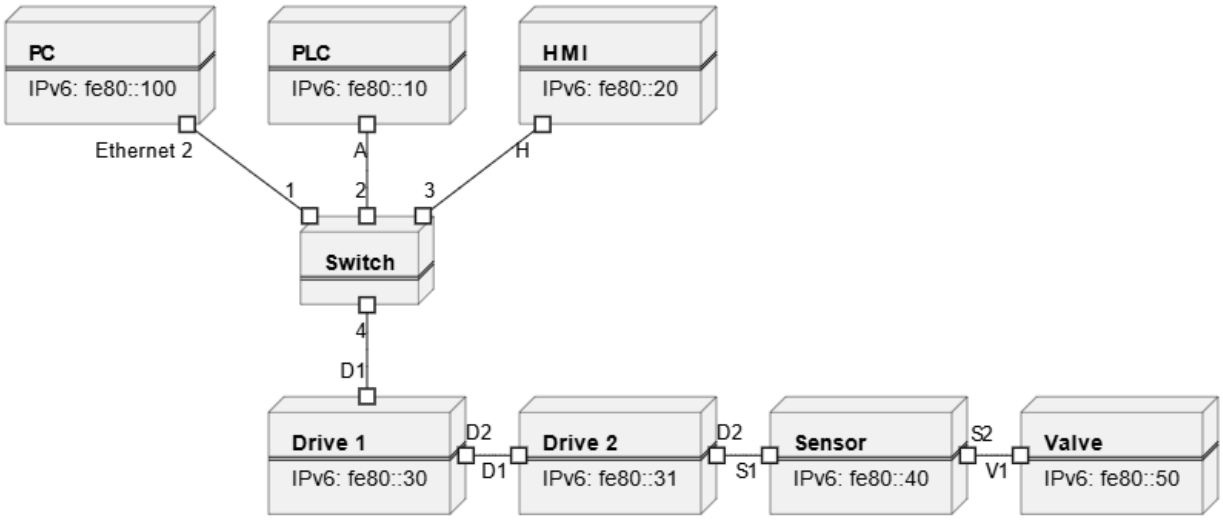


Figure 5 Network configuration on an example machine

Figure 5 defines the desired IPv6 addresses for the devices within the example machine. For example, the PLC has IPv6 link-local address fe80::10. Figure 5 also illustrates the intended physical network connections between all devices, including the network infrastructure devices, such as the Switch. For example, the HMI device must be connected to Switch port 3, the Sensor device port S1 must be connected to Drive 2 port D2 and Sensor device port S2 must be connected to Valve port V1.

The configuration illustrated in Figure 5 can be automatically deployed on a freshly assembled machine. Figure 6 presents the state of the freshly assembled machine shortly after power-up, once SLAAC has assigned IPv6 link-local addresses and the LLDP protocol has completed neighbor discovery.

Classified as Business

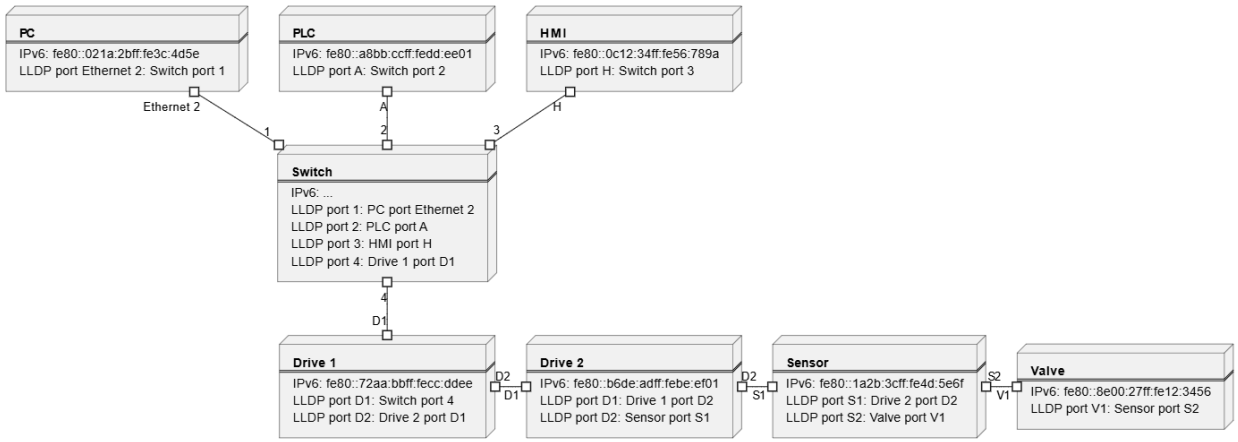


Figure 6 State of freshly assembled machine

Note that all IPv6-capable devices use an automatically determined IPv6 link-local address. Note that LLDP-capable devices have information about neighboring devices for each of their Ethernet ports. The EtherNet/IP and CIP stacks on the devices are operational. The EtherNet/IP devices can be discovered via the ListIdentity Encapsulation Protocol command. The LLDP Data Table Object can be queried for LLDP results.

All the information presented in Figure 6 can be read from the CIP capable devices using standard CIP communication mechanisms. Network infrastructure devices may offer multiple ways to retrieve LLDP results information from them.

The information obtained from the devices in Figure 6 can be compared with the reference model presented in Figure 5 to validate the correctness of the physical connections between devices. This step helps identify potential miswiring issues, which can be costly and time-consuming to diagnose, particularly in large-scale installations.

Once the physical connection checks have passed, the network configuration defined in Figure 5 can be commissioned onto the newly assembled machine. The reference model associates devices configuration with their positions within the topology. The standard CIP communication mechanisms can be used to configure the devices on the freshly assembled machine into the desired state.

The mechanism described above can also be applied after the machine commissioning phase, as it also simplifies device replacement scenario. This is particularly valuable because it provides a unified, automated approach to configuring a device that is connected to the network as a replacement for a failed unit. As a result, it reduces the workload for the machine maintenance crew by eliminating the need to understand details of the IP network. It also removes the requirement to preconfigure IP addresses before connecting a new device to the network. Figure 7 illustrates the state of the machine shortly after the previously failed device - the Sensor - has been replaced with a new unit.

Classified as Business

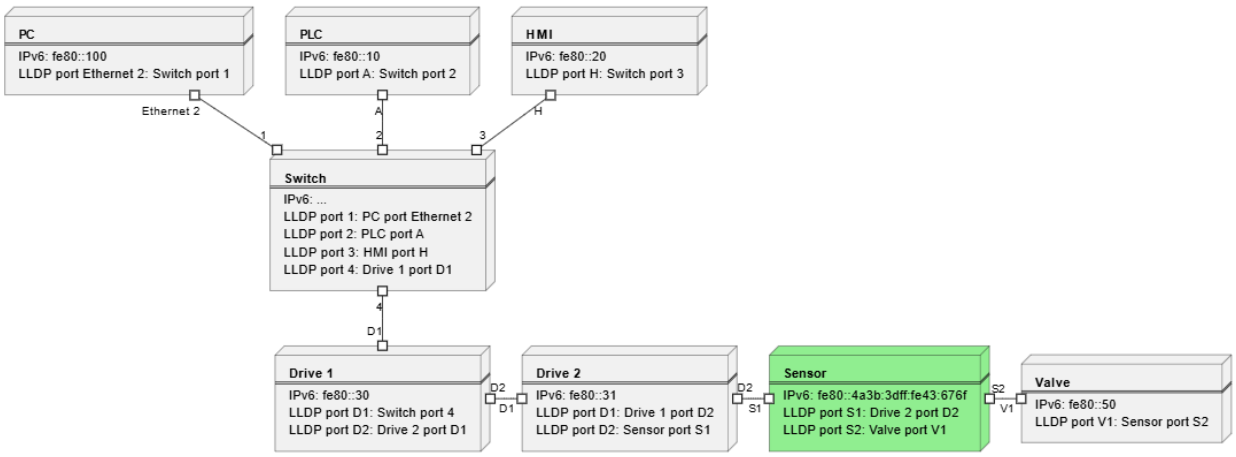


Figure 7 State of the machine after Sensor replacement

Note that the newly connected Sensor device has automatically determined an IPv6 link-local address. LLDP results are available for the Sensor device and its neighbors. The newly connected device can be validated for aspects such as device type, device revision, device capabilities and other criteria. If all checks are passed, the reference configuration can be applied. The configuration procedure is not limited to the setting of IPv6 link-local address. This procedure can also include activities such as updating the firmware revision of the device and setting multiple device parameters, for example IANA Port Admin configuration, IANA Protocol Admin configuration, Ethernet speed value and others. The device configuration can be achieved using existing CIP functionality.

Device provisioning with IPv4

The device provisioning mechanism described in the Device Provisioning with IPv6 chapter can also be implemented using IPv4. To achieve that, the IPv6 capability to automatically generate a link-local address - an essential pilar of device provisioning solution - must be realized using mechanisms available within IPv4. Possible approaches include the use of BOOTP, DHCP, and IPv4 APIPA. The IPv4 CIP devices may selectively support BOOTP and DHCP, as these features are not mandatory within the EtherNet/IP specification. The applicability of IPv4 APIPA has not yet been explored in the context of EtherNet/IP.

Summary

The device provisioning approach presented in this paper reduces deployment complexity, enhances interoperability, and enables advanced commissioning workflows. As a result, it represents a key enabler for scalable, reliable, and user-friendly system integration in EtherNet/IP networks.

While automatic device provisioning is technically feasible with IPv4, an IPv6-based approach using SLAAC is inherently more seamless. SLAAC is a fundamental IPv6 capability and IPv6 CIP specification enhancements propose it as a mandatory component of IPv6 EtherNet/IP. Support for automatic device provisioning with IPv4 may become available in EtherNet/IP in the future.

Hierarchical Autoconfiguration with Names

This section describes one possible method for building self-configuring hierarchies of industrial devices in a converged IT/OT plant network using open-standard protocols, open-source DHCP and DNS stacks,

EtherNet/IP and CIP. Note that no SEs were proposed yet to cover DHCP and DNS stack configuration of Router-Tier Gateway and Devices, nor technology requirements for them.

After that, the device:

1. Automatically receives a GUA or ULA IPv6 address from the correct hierarchical prefix.
2. Gets configured per the methods mentioned in the Device provisioning with IPv6 section.
3. Receives FQDN, whose DNS suffix reflects its position within the plant.

Devices nest, for example:

1. Field devices are aggregated to
2. Machines, which are aggregated to
3. Lines which are aggregated to
4. Floors, which are aggregated to
5. Plants living in an IPv6 private scope (here ULA example is used)

The example nested hierarchy is presented in Figure 6.

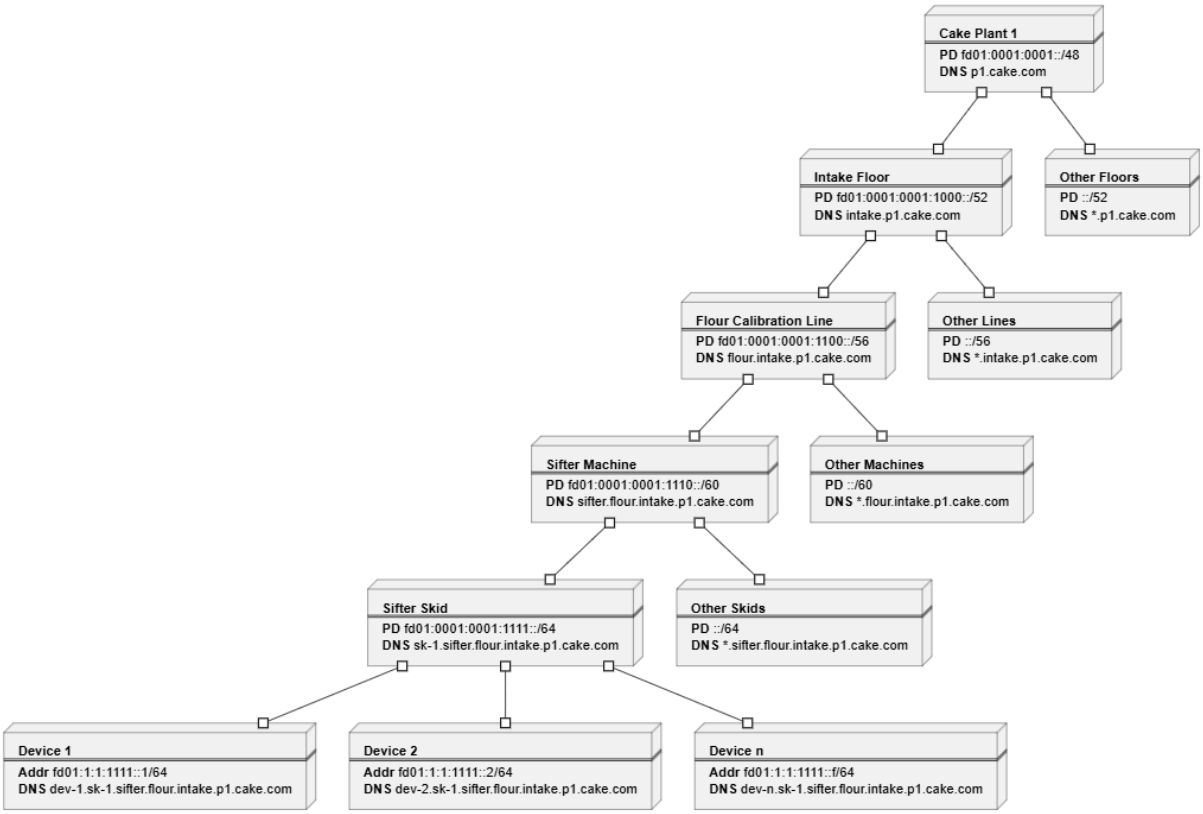


Figure 6 Nesting of a Plant Network

To achieve this, two hierarchies need to be established: An addressing hierarchy built by cascaded DHCPv6 Prefix Delegation (DHCPv6-PD) and a naming hierarchy built with delegated DNS zones fed by Dynamic DNS (DDNS), where local uniqueness is established on the link with mDNS.

Nodes in this hierarchy all use LLDP exposed and configured through EtherNet/IP and CIP, to discover where the devices physically sit as discussed in section Device provisioning with IPv6,yielding

Classified as Business

configuration by location. Replacing a device on a given port restores the same address and FQDN automatically.

Problem statement

Modern plants are converging IT and OT into a common IP fabric. EtherNet/IP and CIP carried over standard Ethernet and TCP/UDP is the dominant industrial application protocol in this space. Historically, OT addressing has been static, manual and flat with heavy use of Layer 2 NAT for nesting.

This does not scale to large, deeply nested installations, and it does not exploit IPv6's abundance of address space or its native autoconfiguration features.

Solution goal

Instead of NAT nesting and static, manual configuration, this solution strives to do prove there is another way: Connect a correct device (catalog number) to the port it should be connected to, and everything is automatically taken care of.

With this action, the device will obtain a temporary auto-configured address that is unique in the system and can be used for communication. Subsequently, the system notices a new device on a port it knows, after verification of the device compatibility, it provides it with a hostname. The hostname is translated to the FQDN by the device asking its router for a suffix. The device then obtains its ULA or GUA address, either by DHCPv6 so that the gateway can register it in DNS, or by composing it from the RA it sees on the network.

Definition of terms

First it is needed to define terms used throughout this section and example configurations.

Device

A device is one physical device with physical Ethernet port or ports on it. It carries the host name information supplied to it by the user or configured during topology-based configuration process. It uses subnet prefix provided by the router-tier gateway. Devices could be End-nodes, CIP Routers or Controllers. It must carry a DHCPv6 client for IA_NA (address) and the Client FQDN option (Option 39), an mDNS querier and responder, an LLDP agent and a CIP stack. LLDP can be presented using the LLDP object or SNMP MIB, if it contains more ports. It does not delegate.

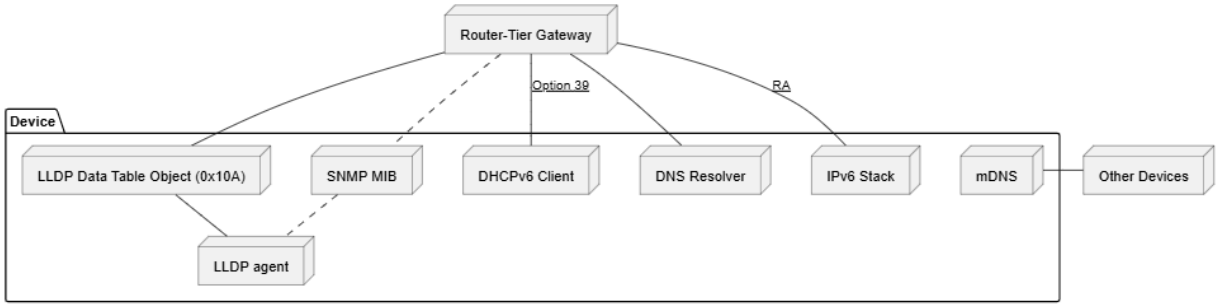


Figure 8 Device Components

Router-Tier Gateway

Classified as Business

A Router-tier Gateway is a device capable of DHCPv6 client features for getting an upstream prefix, it is capable of DHCPv6 server features to delegate its own pool of prefixes, and it is capable of DDNS management for its zone. This device is important for machine nesting as it delegates IPv6 subnet prefixes and is configured to carry correct DNS suffixes, which are then merged with End-node host names. Every subtree must have one. This device could also watch over the segment and carry the configuration of the devices based on their physical location.

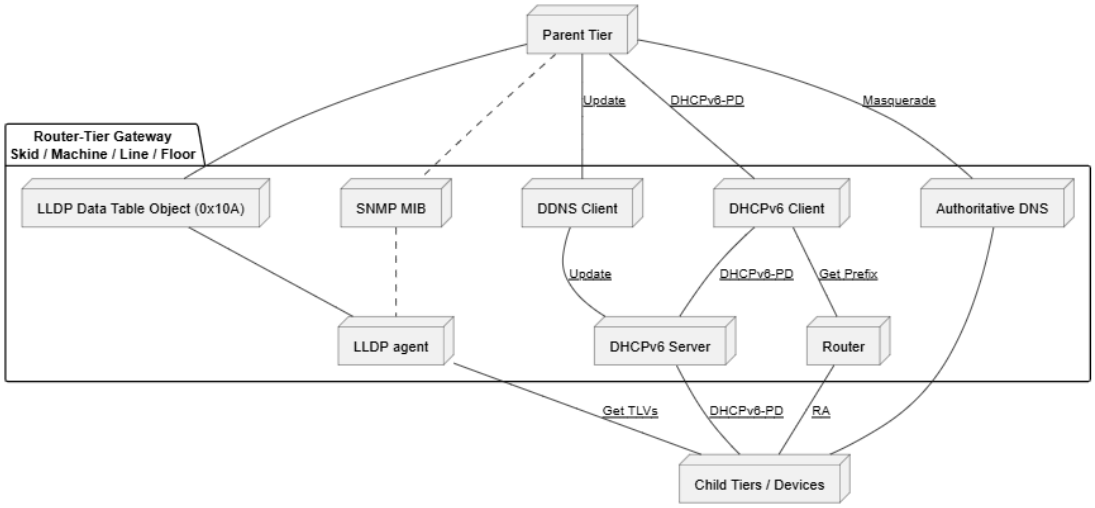


Figure 9 Router-Tier Gateway Components

Machine

A machine here means a network of devices grouped together for a specific purpose. It contains devices, other machines or skids and is connected to upper tiers via a Router-Tier Gateway.

Skid

A skid is a machine to be reused in other machines, i.e. a sub-machine.

Background and Standards

This section uses following standard open and proprietary technologies for addressing, name resolution and topology discovery.

IPv6 addressing

Unique Local Addresses (RFC 4193) [7]: the fc00::/7 private scope. The plant's root prefix then could partition the remaining portion of the 64-bit subnet prefix address element.

DHCPv6-PD (RFC 9915) [8]: a Requesting Router asks an upstream Delegating Router for a prefix (not a single address). The requestor then sub-delegates smaller prefixes to its own clients.

ND (RFC 4861) [9], **SLAAC (RFC 4862)** [10].

Name resolution

mDNS (RFC 6762) [11]: link-local name claim and service discovery via the probe/announce mechanism. mDNS gives uniqueness on one link, hence it assures uniqueness of the host name part.

DNS delegation: each tier owns an authoritative zone that is NS-delegated by its parent. The naming tree is therefore literally the plant tree.

DDNS (RFC 2136) [12], potentially secured by TSIG (RFC 8945) [13] or GSS-TSIG (RFC 3645) [14]: the DHCP authority writes AAAA/PTR records on the device's behalf. The DHCPv6 Client FQDN option (Option 39, RFC 4704 [15]) carries the device's requested short name into the exchange.

Topology discovery

LLDP (IEEE 802.1AB) [16]: each device periodically advertises a set of TLVs to its directly connected neighbors: Chassis ID, Port ID, TTL, System Name, System Description, System Capabilities, Management Address, Port Description, plus organizationally specific TLVs.

LLDP is single-hop and therefore an authoritative source of physical adjacency: who is plugged into which port. The System Capabilities TLV distinguishes a router-tier gateway (Bridge + Router bits) from a leaf station (Station Only bit). The Management Address TLV can carry the parent's IPv6 management address, telling a child exactly where to direct DHCPv6/DNS.

Configuration Methods

This solution uses following configuration and diagnostic methods:

- Configuration files of standard DHCP and DNS stacks
- EtherNet/IP and CIP
- SNMP

EtherNet/IP and CIP

Table 4 EtherNet/IP and CIP Configuration Methods

Object	Purpose
Identity	Provides device identification and assures that the device on the physical port is an expected one.
IPv6 Link-Local Object	Allows configuration and diagnostic of link-local IPv6 address.
IPv6 Global-Unicast Object	Allows configuration and diagnostic of non-locally-scoped addresses, such as Unique Local Addresses (private addresses), Global Unicast Addresses.
LLDP Data Table object	Provides neighboring LLDP TLVs for physical device localization.

SNMP

For devices not supporting the LLDP Data Table object, the SNMP OID can be used to get the same localization information, namely lldpRemTable OID: 1.0.8802.1.1.2.1.4.1 [17].

Architecture

Now the terms, standards and configuration methods used are established, the focus can move to the architecture of the solution.

Private network range fc00::/7

The cornerstone of the architecture is the private network address range fc00::/7. This range does not route outside the private network, which is a fit for isolated OT plant networks. Global unicast range can also be used, but then the network privacy would need to be assured using IT methods, for example firewalls.

For subnets within factories that contain machines that can nest as shown in the Figure 6, address spaces must be claimed. In IPv6, the subnet fc00::/7 is reserved for local addressing. It is similar to IPv4's range 10.0.0.0/8. This network can be used by OT departments to partition further, leaving 57 bits to partition. This comes with several caveats:

1. Typically, fd00::/8 is used as fc00::/8 is not a defined space shortening the partition to 56 bits.
2. If VPN is used, these networks can overlap on one machine when connected to both private networks with the same prefix. To mitigate this, some IPv6 implementations established the principles of ULA, with 40 randomly generated bits to prevent routing clashes. This shortens the partition further to 16 bits leaving 65536 subnets to partition for one ULA prefix.

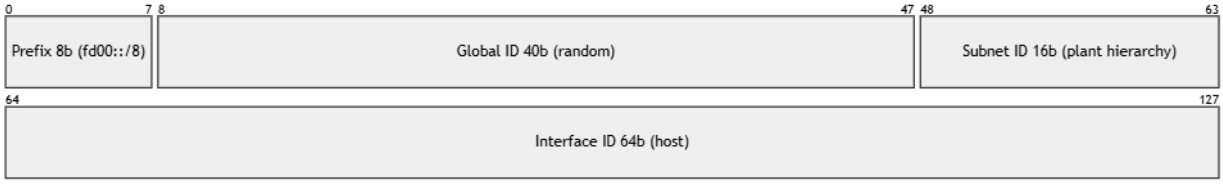


Figure 10 ULA Structure

In the scenario described earlier, the factory OT department does not know beforehand what machines are going to connect as they are built by OEMs. It is important to put constraints on OEM machines, so that dynamic nesting is possible without the need to introduce NAT.

To increase the partitioning space for a plant, it is possible to use multiple ULAs, or partition more of the private network range space, however, VPN collisions would need to be resolved. These are the same as IPv4 VPN subnet prefix collisions happening with today's network, where the solutions may be NAT66, or using a workstation GUA for remote access instead of a VPN.

OEM Constraints

For nesting to work for numbers and names and for it to allow subnet isolation without the need for NAT, constraints must be defined for OEMs and Machine Builders and the OT department of a factory.

1. OEMs shall not use ULA or GUA inside their machines

The machines should work only with link-local addresses and put no hard-coded address values within them. Machines shall be either:

- Fully passive or,
- allow configuration of the addresses of uplink dependencies in their controller parameters,

Classified as Business

- or use DNS (mDNS) for resolution of uplink dependencies.

2. OEMs shall document required partition of a subnet prefix if they contain nested machines

Machines containing other machines must specify the size of the subnet prefix partition they wish to configure. The size must be as small as possible, only as large as the machine needs for the sub-machines to be addressed. As discussed in section Example Numbering Scheme, in order for Reverse DNS to work, the smallest real size is a nibble.

3. **OEMs devices shall either provision the machine with a Router-Tier Gateway or allow devices to listen to RA to obtain their GUA or ULA prefix**

It is vital for the factory OT department to be able to configure and update all devices and to gather diagnostics. If a Router-Tier Gateway is a part of the machine, DHCPv6-PD must be used by the machine's Router-Tier Gateway to accept the partition provisioned for it by an upper-level Router-Tier Gateway and ultimately the factory OT department.

Device Vendor Constraints

1. Device vendors must allow link-local setting and also ULA/GUA address

In environments without NAT, every device must be a part of link-local scope for control, and a global/private scope for asset management.

Example Numbering Scheme

Given previous constraints and the proposed design in Figure 6, the following addressing scheme is possible, presuming ULA is used for the plant:

Table 5 Plant addressing scheme

Tier	Prefix	Bits gained	Fan-out	Cumulative capacity
Plant	/48	-	1 root	-
Floor	/52	4	16 floors	16
Line	/56	4	16 lines / floor	256
Machine	/60	4	16 machines / line	4,096
Skid	/64	4	16 skids / machine	65,536
Device	-	64	2^{64} devices / skid	2^{80}

Reflecting on previous constraints, the vendors of the skids would have no space to partition and use only link-local addresses, machines could contain up to 16 skids or sub-machines, lines could accommodate up to 16 machines and floors could accommodate up to 16 lines. The partitioning can be changed depending on the plant setup, so for example floors and lines might collapse into 256 lines as these are fully in the hands of plant's OT department, at a cost of physical location not being reflected in the address, or more private network can be addressed as discussed in the section Private network range fc00::/7.

WARNING: It is recommended to never partition the prefixes to anything else than multiples of 4 as reverse DNS for IPv6 is nibble-aligned.

Example Naming Scheme

The example naming hierarchy is congruent with the addressing hierarchy: the DNS zone tree is the prefix tree. Each tier owns one authoritative zone, NS-delegated by its parent, and appends its own label to whatever the child below proposes. Walking down the tree therefore both shortens the prefix and lengthens the FQDN by exactly one component per hop.

Forward naming

Reusing the addressing example (fd01:0001:0001::/48, DNS root p1.cake.com), the forward zones nest one-to-one with the prefixes:

Table 6 Plant naming scheme

Tier	Prefix	Owned forward zone	Delegated by
Plant	`/48`	`p1.cake.com`	- (root)
Floor	`/52`	`intake.local (mDNS)` `intake.p1.cake.com`	Plant
Line	`/56`	`flour.local (mDNS)` `flour.intake.p1.cake.com`	Floor
Machine	`/60`	`sifter.local (mDNS)` `sifter.flour.intake.p1.cake.com`	Line
Skid	`/64`	`sk-1.local (mDNS)` `sk-1.sifter.flour.intake.p1.cake.com`	Machine
Device	`host`	`dev-2.local (mDNS)` `dev-2.sk-1.sifter.flour.intake.p1.cake.com`	Skid / Machine

A leaf device that proposes only the short label dev-2 therefore ends up as `dev-2.sk-1.sifter.flour.intake.p1.cake.com`, where the suffix is contributed entirely by the tiers above it, never invented by the device. The device also presents itself as `dev-2.local` to the other devices on the same local link.

The construction obeys three rules:

1. Authority-assigned suffix. The device proposes only a short label via DHCPv6 Option 39. Then the Router-Tier Gateway appends its own zone and returns the result.
2. Link-local uniqueness. mDNS collision detection guarantees the label is unique on its link.
3. Plant-global uniqueness. The DNS-suffix hierarchy gives uniqueness across links. The combination of unique local names and DNS-suffix uniqueness makes short name assignments safe at plant scale (similarly to what subnet prefix and interface ID do for IPv6 stateless numeric uniqueness).

Reverse naming (PTR)

Reverse DNS answers the inverse question: “Given an address, what is its name?” And it follows the exact same delegation tree, only expressed in the special ip6.arpa domain.

This is especially useful for diagnostics, when only an IP address is known. If reverse naming is employed, just pinging the address can yield its FQDN, so that people diagnosing an issue immediately identify and locate the device.

An IPv6 address is reversed one nibble (4 bits) at a time, least-significant first, with a dot between every nibble, suffixed by ip6.arpa. Because each tier boundary in the addressing plan is nibble-aligned, every prefix length maps to a clean reverse-zone cut:

Classified as Business

Table 7 Reverse naming scheme

Tier	Prefix	Reverse zone (ip6.arpa)
Plant	fd01:0001:0001::/48	1.0.0.0.1.0.0.0.1.0.d.f.ip6.arpa
Floor	fd01:0001:0001:1000::/52	1.1.0.0.0.1.0.0.0.1.0.d.f.ip6.arpa
Line	fd01:0001:0001:1100::/56	1.1.1.0.0.0.1.0.0.0.1.0.d.f.ip6.arpa
Machine	fd01:0001:0001:1110::/60	1.1.1.1.0.0.0.1.0.0.0.1.0.d.f.ip6.arpa
Skid	fd01:0001:0001:1111::/64	1.1.1.1.1.0.0.0.1.0.0.0.1.0.d.f.ip6.arpa

The plant owns the top reverse zone and NS-delegates each /52, /56, /60, /64 reverse zone down the tree exactly where it delegates the matching forward zone. PTR registration is then a purely local DDNS action at the owning tier: when a tier server hands out an address, it writes both the forward AAAA and the reverse PTR into its own two zones in one transaction.

A full forward/reverse round-trip for the worked device looks like this:

```
AAAA  dev-2.sk-1.sifter.flour.intake.p1.cake.com fd01:1:1:1111::2/64
PTR   2.0.0.0.1.1.1.1.1.0.0.0.1.0.0.0.1.0.d.f.ip6.arpa dev-2.sk-
1.sifter.flour.intake.p1.cake.com
Maps :0002___:1111___:0001___:0001___:fd01
```

If those two records resolve to each other, the device's name and its position in the plant are provably consistent.

Protocol Flows

This section describes the full process of connecting a correct Device or a Router-Tier Gateway to a correct port and what happens, so that the system keeps itself configured and named at all times.

Device onboarding / Device replacement

A device (dev-2) is connected to a machine's Router-Tier Gateway's /64 link.

Classified as Business

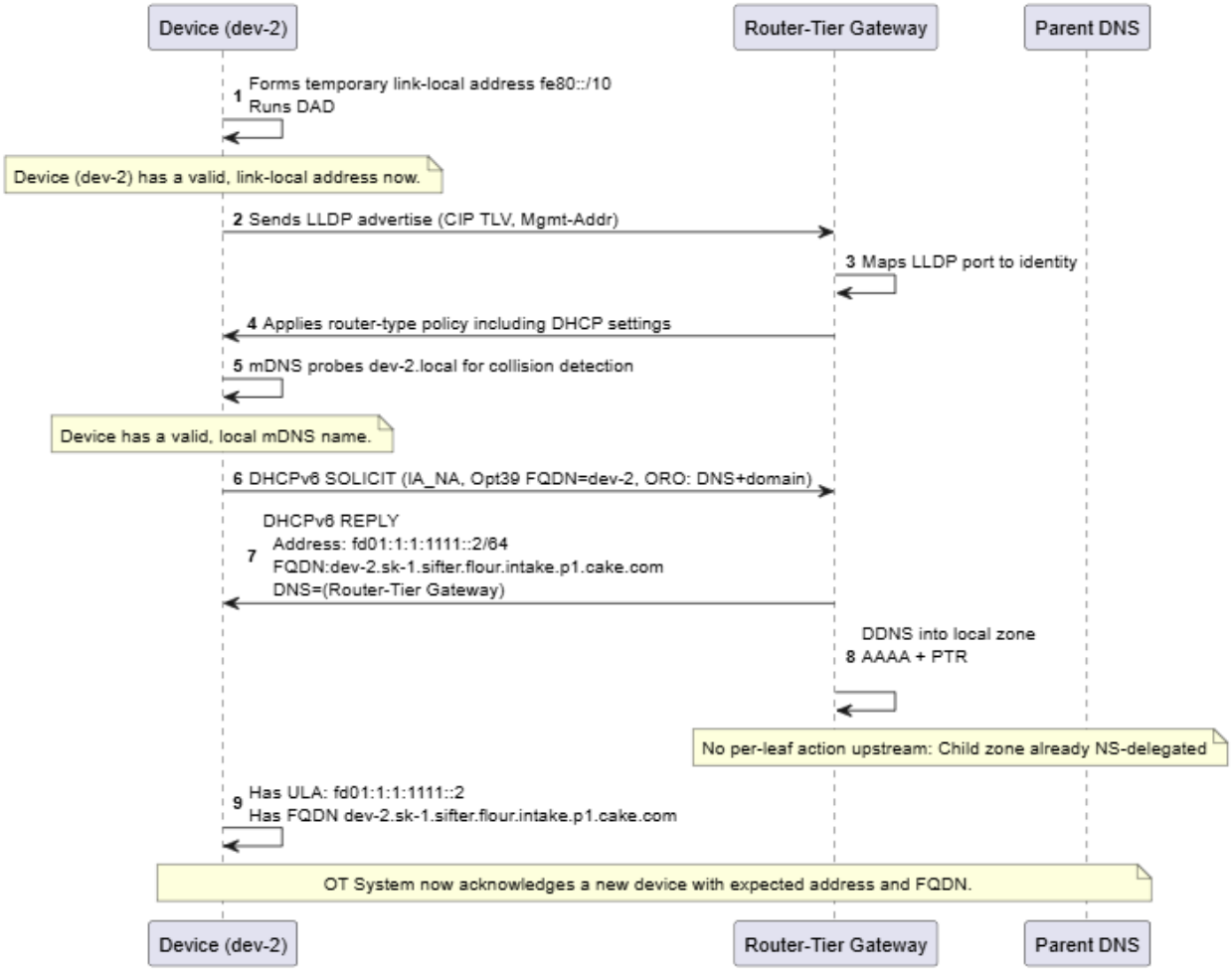


Figure 11 Device Onboarding and Replacement

Key points:

- The device requests IA_NA (an address), not a prefix, so the DHCPv6 server knows the address and can register DNS for it.
    - If the device address is configured manually, the device itself would need to update the zone authority (Router-Tier Gateway) on its own, therefore DDNS clients on devices would need to be mandatory.
- The machine gateway only owns that machine's /64s, so anything it hands out is automatically in the right prefix.
- mDNS assures no collisions within the prefix.
- The suffix is provided by the gateway from its own zone, the device never invents its domain.

Router-tier onboarding and sub-delegation

A new machine gateway is connected to a line gateway. The machine must obtain a prefix, then become a server/DNS authority for everything beneath it.

Classified as Business

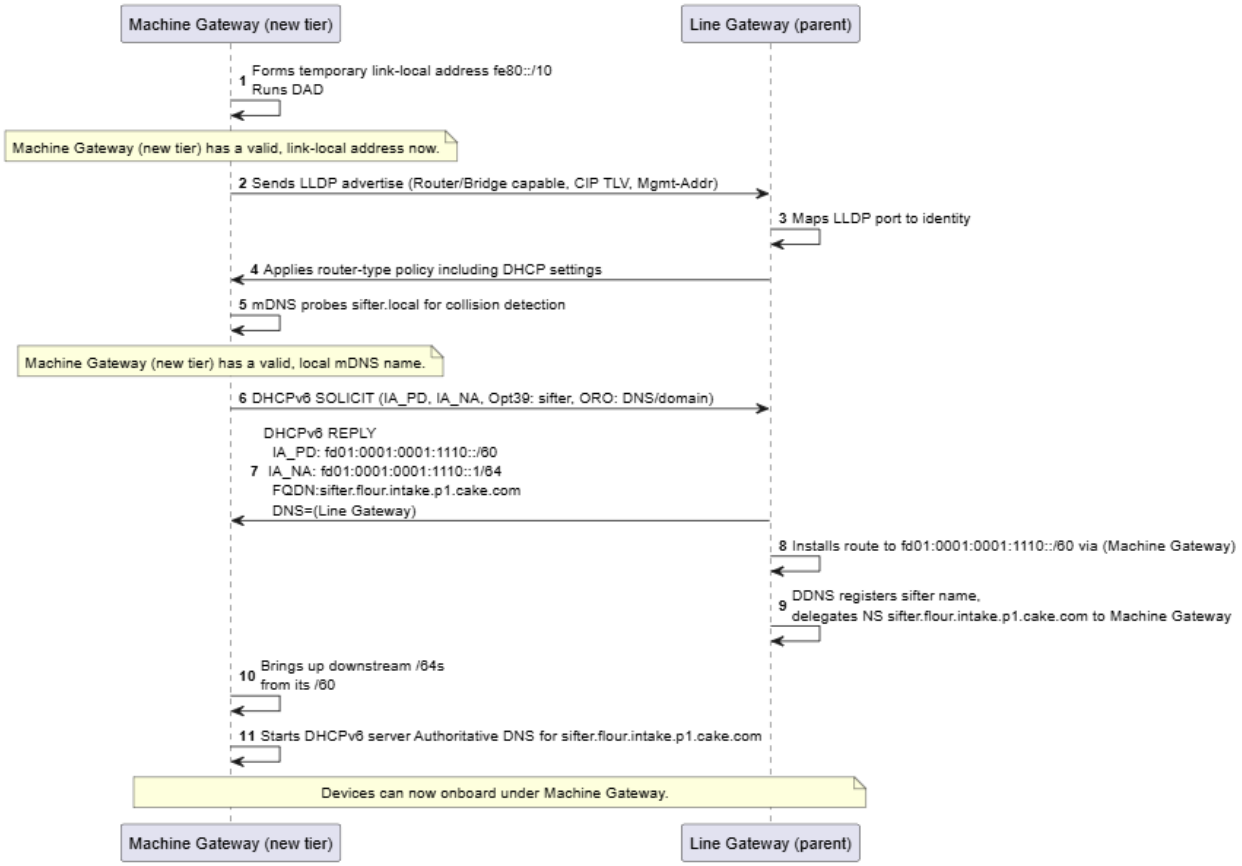


Figure 12 Router-Tier Onboarding and sub-delegation

This flow presumes that routes follow lease**.** The /60 route in the parent is installed/withdrawn with the PD lease. With strict aggregation, this single route is sufficient.

Demo using standardized protocols and FOSS

You can find the demo on https://github.com/EtherNet-IP-Unofficial/jkb-naming-hierarchy

Conclusion

As it was shown, with a careful network design simply connecting a device of compatible identity alone can be sufficient for it to get a correct, hierarchical IPv6 address and FQDN, with arbitrary nesting.

As the demo showed, it is achievable today with standardized protocols and FOSS. The construction is two congruent hierarchies: DHCPv6-PD for nested prefixes and delegated DDNS zones for nested names, joined by mDNS for link-local name claims.

The demo setup is only achievable if pre-requisite constraints are met, therefore a decision must be made on mandating them or on adding device profiles so that the customers can build similar setups with all devices or devices capable of it, respectively.

CIP Specification enhancements to support presented functionality with recommended protocol stack and supporting technologies and also with object support for configuration and diagnostics will follow.

IPv6 Future in Industrial Automation

One of the key factors behind the success of EtherNet/IP is that it leverages the same technologies that are used in standard IT networks. EtherNet/IP is based on widely adopted protocols such as Ethernet, IP, TCP and UDP and it can be deployed on the same infrastructure as conventional computer networks. This makes EtherNet/IP straightforward to integrate with existing IT environments.

The IT technologies and infrastructure evolve rapidly while industrial automation domain is inherently conservative, favoring well-proven and reliable solutions. Historically, new technologies have been adopted in industrial automation with a noticeable delay. An example of that is IPv4 based EtherNet/IP which was first standardized in 2001, while at that time IPv4 based Word Wide Web reached approximately half a billion users. Over the years, EtherNet/IP has consistently incorporated technologies that were first validated in standard networking environments. Examples include QoS, LLDP, RSTP, SNMP, and PRP. These capabilities are often embedded in CIP devices and can be managed through dedicated CIP interfaces.

This conservative approach within industrial automation is largely driven by end customers. The industrial automation solutions introduced decades ago often remain the standard in production environments today. Only a relatively small group of customers actively drives change and articulates the need for new capabilities.

Currently, the most demanding customers are increasingly requesting IPv6. Their needs are driven by several factors:

- The growing scale of their systems, where the limitations of IPv4 become a significant constraint.
- The requirement to integrate seamlessly with enterprise IT infrastructures that are already based on IPv6.
- The ever-growing need to extract and transfer data from the factory floor to enterprise systems for traceability and for turning raw machine data into business value via data driven manufacturing optimization.
- Governments mandate IPv6 on network equipment.

IPv6 today is a mature technology, as of today approximately 50% of global internet traffic is carried over IPv6. Today IPv6 support is being incorporated into the EtherNet/IP specification. Its adoption in industrial automation is expected to follow a gradual transition similar to that observed in other domains. Initially, existing IPv4 devices will be enhanced with IPv6 capabilities, resulting in dual-stack solutions that allow coexistence with current installations while enabling interoperability with IPv6-based infrastructure. Over time, IPv6 is expected to become the preferred option, ultimately leading to IPv6-only deployments, in line with modern IT network designs used in new facilities today.

Summary

This paper presents the current state and direction of introducing IPv6 support into CIP and EtherNet/IP. It can be viewed as an update on the progress of the IPv6 roadmap defined in [2], with a particular focus on the implementation details of Phase 0.

The paper outlines the key challenges associated with incorporating IPv6 into the CIP object model and provides justification for the proposed solutions. These include the introduction of the IPv6 Link-Local Interface Object and IPv6 Global Unicast Interface Object, as well as a new EtherNet/IP Interface Object. The proposed models enable IPv6 capabilities in a manner that maximizes reuse of existing IPv4 solutions while ensuring backward compatibility.

For CIP Safety, the transition to IPv6 introduces challenges due to the mismatch between existing SNN structure and IPv6 addressing. Three solution concepts were analyzed, with an SSN decoupled from IP address model emerging as the preferred direction. This approach maintains backward compatibility while aligning with safety principles and supporting future network architectures.

The paper further demonstrates that IPv6 significantly simplifies device provisioning and device-replacement workflows. The combination of SLAAC and LLDP enables automated, topology-aware commissioning and configuration, and also substantially reduces the complexity of replacement procedures.

Hierarchical autoconfiguration using IPv6 prefixes and DNS delegation is also introduced, illustrating how industrial networks can scale using structured addressing and naming aligned with plant topology. This approach eliminates the need for manual addressing schemes and supports dynamic system expansion.

Finally, the paper explores the anticipated evolution of IPv6 within industrial automation.

Acknowledgements

The authors would like to acknowledge the valuable contributions of Todd Wiese, Jim Grosskreuz, Steven Seidlitz, and Nick Stay of Rockwell Automation, as well as David Crane of ODVA, for their contributions to the IPv6 Safety concept selection effort.

References

[1] U. G. S. Administration, "Internet Protocol Version 6 (IPv6) Policy," 2025. [Online]. Available: https://www.gsa.gov/directives-library/internet-protocol-version-6-ipv6-policy-1.

[2] J. Korbel, B. Batke and F. Zembok, "An IPv6 Roadmap for EtherNet/IP," in *2025 Industry Conference & 23rd Annual Meeting*, Clearwater Beach, 2025.

[3] D. XU, Y. YU, P. Brooks and B. Batke, "EtherNet/IP over IPv6 - Evolution, not Revolution for the World's Leading Industrial Ethernet Variant," in *2012 ODVA Industry Conference & 15th Annual Meeting*, Stone Mountain, 2012.

[4] ODVA, The CIP Networks Library, Volume 2: EtherNet/IP Adaptation of CIP, Ann Arbor: ODVA, Inc., 1999-2026.

[5] F. Gont, ""A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)," 2014. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7217.html. [Accessed July 2026].

[6] T. Narten, R. Draves and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," 2007. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4941. [Accessed July 2026].

[7] R. Hinden and B. Haberman, "Unique Local IPv6 Unicast Addresses," October 2005. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4193. [Accessed 1 July 2026].

[8] T. Mrugalski, B. Volz, M. Richardson, S. Jiang and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," January 2026. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc9915. [Accessed 1 July 2026].

[9] T. Narten, E. Nordmark, W. Simpson and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," September 2007. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4861. [Accessed 1 July 2026].

[10] S. Thomson, T. Narten and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," September 2007. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4862. [Accessed 1 July 2026].

[11] S. Cheshire and M. Krochmal, "Multicast DNS," February 2013. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6762. [Accessed 1 July 2026].

[12] P. Vixie, S. Thomson, Y. Rekhter and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," April 1997. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2136. [Accessed 1 July 2026].

Classified as Business

[13] F. Dupont, S. Morris, P. Vixie, D. Eastlake 3rd, O. Gudmundsson and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)," November 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8945. [Accessed 1 July 2026].

[14] S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead and R. Hall, "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)," October 2003. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc3645. [Accessed 1 July 2026].

[15] B. Volz, "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option," October 2006. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4704. [Accessed 1 July 2026].

[16] Institute of Electrical and Electronics Engineers, Inc., Station and Media Access Control Connectivity Discovery (IEEE Std 802.1AB-2009), New York: IEEE, 2009.

[17] oidref.com, "Reference record for OID 1.0.8802.1.1.2.1.4.1.1," [Online]. Available: https://oidref.com/1.0.8802.1.1.2.1.4.1.1. [Accessed 1 July 2026].

[18] ODVA, The CIP Networks Library, Volume 1: Common Indutstrial Protocol, Ann Arbor: ODVA, Inc., 2001-2026.

[19] ODVA, The CIP Networks Library, Volume 5: CIP Safety, Ann Arbor: ODVA, Inc., 2005-2026.

****

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use. Copyright ©2026 ODVA, Inc. All rights reserved. For permission to reproduce excerpts of this material, with appropriate attribution to the author(s), please contact ODVA on: TEL +1 734-975-8840 FAX +1 734-922-0027 EMAIL odva@odva.org WEB www.odva.org. CIP, Common Industrial Protocol, CIP Energy, CIP Motion, CIP Safety, CIP Sync, CIP Security, CompoNet, ControlNet, DeviceNet, and EtherNet/IP are trademarks of ODVA, Inc. All other trademarks are property of their respective owners.