Moving EtherNet/IP towards “Plug and Play”

Jamin D. Wendorf
Real Time Automation

Technical Track

www.odva.org
**Terminology**

**Dynamic Host Configuration Protocol (DHCP)** – an automatic configuration protocol used to assign an IP address to a DHCP client.

**Electronic Data Sheet (EDS)** - text based configuration file used during network commissioning.

**Plug and Play (PnP)** – configuring devices without user intervention.
**Terminology**

**CIPSE** – Common Industrial Protocol Specification Enhancement (i.e. an enhancement proposal to Volume 1 of the ODVA CIP Network Library)

**O→T** – Originator to Target (message direction from originator (scanner) to target (adapter))

**T→O** – Target to Originator (message direction from target (adapter) to originator (scanner))
Out-of-the-box to Data Exchange

- Wire up device (not in presentation)
- Get IP Address (default DHCP)
- Verify necessary configuration
- Establish communications
- Parse data (not in presentation)
Establishing a connection: DeviceNet

- **CAN packet**: 8 data bytes
- **Multiple messages needed to establish a connection**
  - Allocate
  - Configure produce and consume instances
  - Read multiple parameters to identify default configuration
  - Configure additional parameters (as needed)
  - Verify connection sizes
  - Write EPR to transition to established
Establishing a connection: EtherNet/IP

- Ethernet packet: multiple data bytes (<1500)
- Single message needed to establish a connection
- What’s in the message?
Forward Open Contents

- O→T & T→O Connection IDs
- Connection Serial Number
- Originator’s Vendor ID and Serial Number
- O→T & T→O Requested Packet Intervals and Connection Timeout Multiplier
- O→T & T→O Network Connection Parameters (Connection Size, Type and Ownership)
- Transport Class and Trigger (Cyclic, Change of State, Application Trigger)
- Connection Path
Forward Open Advantages

• A single request
• Configuration data is optionally passed aiding in multiple parameter configuration
• Automatic device replacement
Forward Open Disadvantages

• Multiple parameters for the user to configure incorrectly
  • Invalid RPI
  • Incorrect connection paths (input/output/configuration)
  • Incorrect connection sizes (input/output/configuration)
  • Incorrect configuration data content
  • Missing configuration data
What are the issues?

1. Not a common method to present user with valid combinations for the I/O connection
2. Too many parameters in Forward Open
3. Conformance and Plug Fest Testing
Presenting options to users

No common way to present the user with the valid combinations

- **Embed known adapter devices in scanner configuration tool**
  - Must be a known device
  - Requires constant updates of scanner configuration tool
Presenting options to users

- Generic input box to solicit the input, output and configuration instances and sizes
  - Flexible and fast
  - Must know all configuration parameters
  - Prone to typos
  - Reliant on vendors to supply complete information
Presenting options to users

- **Parse the EDS file and expose the valid connections**
  - Only as good as the EDS file
  - Not all devices support a complete EDS file
  - Not all scanner configuration tools support parsing EDS files (or parse them incorrectly)
Too many parameters

• All parameters are pushed from the scanner to the adapter
• Multiple opportunities for errors
  • Bad RPI
  • Incorrect instances (pairs) for the I/O connection points
  • Incorrect or missing configuration data
Testing

- No standard method of configuration
- Too much time is wasted learning multiple approaches
- CCO isn’t fully supported; dependent on EDS files
What’s the solution?

- **New CIP object - Target Connection List (TaCL) object (still in CIPSE phase)**
  - Easily verify I/O communications to adapter devices
  - Lists all supported connections (Exclusive Owner, Input Only, Listen Only)
  - Data Exchange connection (loop back of data) to prove proper I/O communications and test performance
  - Known instances to allow for “auto-connect”
  - Optional attributes with parameter ranges to allow for tweaking the connection parameters (like multicast or unicast, RPI ranges, etc...)

- **Changes to CCO object to support “auto-connect”**
TaCL Object overview

- **At least one instance would be required**
  - Instance 1: Default Exclusive Owner Connection
  - Instance 2: Default Input Only Connection
  - Instance 3: Default Listen Only Connection
  - Instance 4: Data Exchange Test
  - Instance 5-9: reserved
  - Instances 10+: user defined
TaCL Object overview

- **Connection Read Service Code (0x4C)**
  - Class or Instance Level service code
  - Transport Class and Trigger (cyclic, change of state, application trigger)
  - Originator to Target Network Parameters (Multicast/Point-to-Point, Size, etc...)
  - Target to Originator Network Parameters (Multicast/Point-to-Point, Size, etc...)
  - Originator to Target RPI
  - Target to Originator RPI
  - Connection Path and Data (optional)
TaCL Object overview

- **Connection Read Service Code (cont...)**
  - Information is copied to the Connection Configuration Object and used to build a Forward Open service code
  - Forward Open is “guaranteed” to succeed
TaCL Object overview

- TaCL Attributes
  - RPI ranges
  - Unicast/Multicast or both
  - Priorities supported (Low, High, Scheduled)
  - Transport Triggers supported (COS/Cyclic)
  - I/O Connection Instance Type (Ex Owner, Input Only, Listen Only)
  - I/O Connection name ("Status & Data", "Data Only")
TaCL usage cases

Case 1: Auto Connect – Exclusive Owner
Case 2: Auto Connect – Data Exchange Test
Case 3: List all supported connections
Case 4: Easily add Listen Only connection
Case 1: Auto Connect – Exclusive Owner

- Used to connect to a target device with the default Exclusive Owner configuration
- This is the typical connection a customer needs during initial commissioning
- Once connection is made, data content is still unknown, but this isn’t usually important in the first stage of network commissioning
Case 2: Auto Connect – Data Exchange Test

- Used to connect to a target device and exchange known data to prove bidirectional communications are working
- Useful in conformance testing
- Useful during Plug Fest testing, especially the system test and performance test
- Only fear is that developers could cheat the performance metrics by adding special code
Case 3: List all supported connections

- Useful with scanner configuration tools
- Exposes all supported connections, including exclusive owner, input only and listen only
- Provides meaningful names for each supported connection to assist the user during the selection process
Case 4: Easily add Listen Only connection

- Used to attach to an existing exclusive owner connection
- Provides existing RPI data for input only and listen only connections to ensure the allocation works on the first attempt
How is “Plug and Play” realized?

- **Adapters**
  - Support for TaCL object with “Auto Connect”
  - Support DHCP

- **Scanners**
  - Support for the CCO object with “Auto Connect”
  - Use List Identity to find devices

- **This only allocates the connections, the content of the I/O data exchange isn’t known from the TaCL object.**
Add’l functionality for interop

- Provide a correctly written EDS file and most desirable, embed in device
- Add a web server with diagnostics (counter, number open connections, etc…). Recommended functionality to be defined by the ODVA EtherNet/IP Implementors Workshop at a later date.
- Follow “Recommended IP Addressing Methods for EtherNet/IP™ Devices”
- Follow “Recommended Functionality for EtherNet/IP Devices”
- Pass ODVA Conformance Testing
- Pass Plugfest Interoperability Testing