

Industrial Communication

Chapter 4: Ethernet Modbus TCP/IP

Ethernet Modbus TCP/IP Features

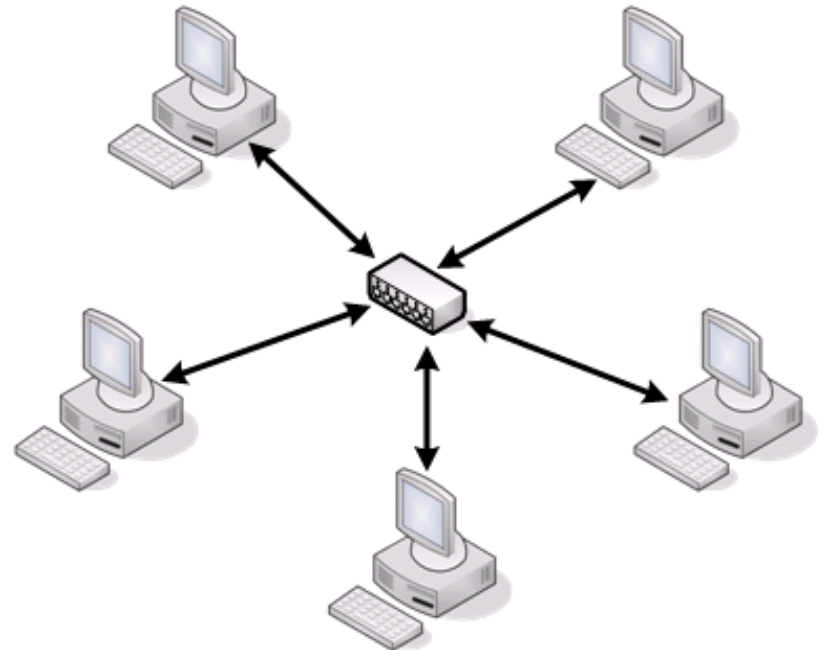
Ethernet Is a Network

- History

- Developed by Xerox PARC (1973 - 1975)
- First standard to be published in 1980 (IEEE 802.3)

- Sharing Information

- Network

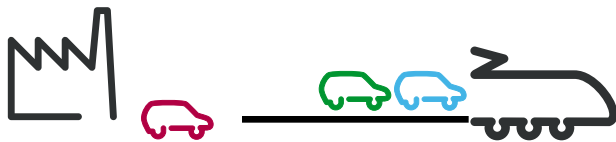


Ethernet Modbus TCP/IP

- Modbus Frames over Ethernet
 - Modbus Serial



- Ethernet Modbus TCP/IP



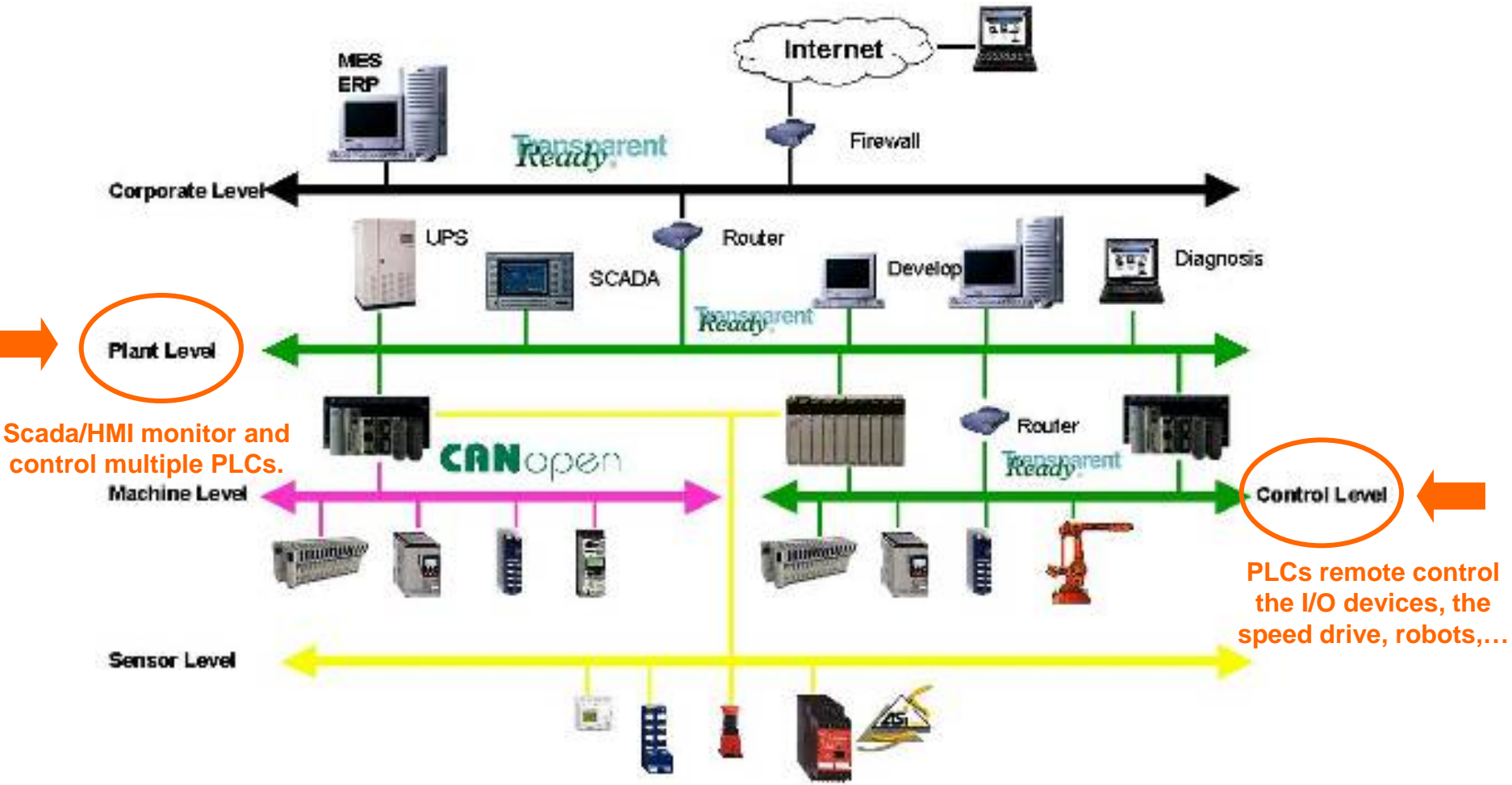
Layers Used Ethernet Modbus TCP/IP

- Osi Model:

APPLICATION	Modbus Protocol
PRESENTATION	<i>Not used</i>
SESSION	<i>Not used</i>
TRANSPORT	Ensure the transfer of the data on the network (TCP/UDP)
NETWORK	Message Routing – IP Protocol
DATALINK	CSMA-CD (random access with collisions)
PHYSICAL	Defines physical media: RJ45, Fiber Optic, Radio, ...

Industrial Ethernet

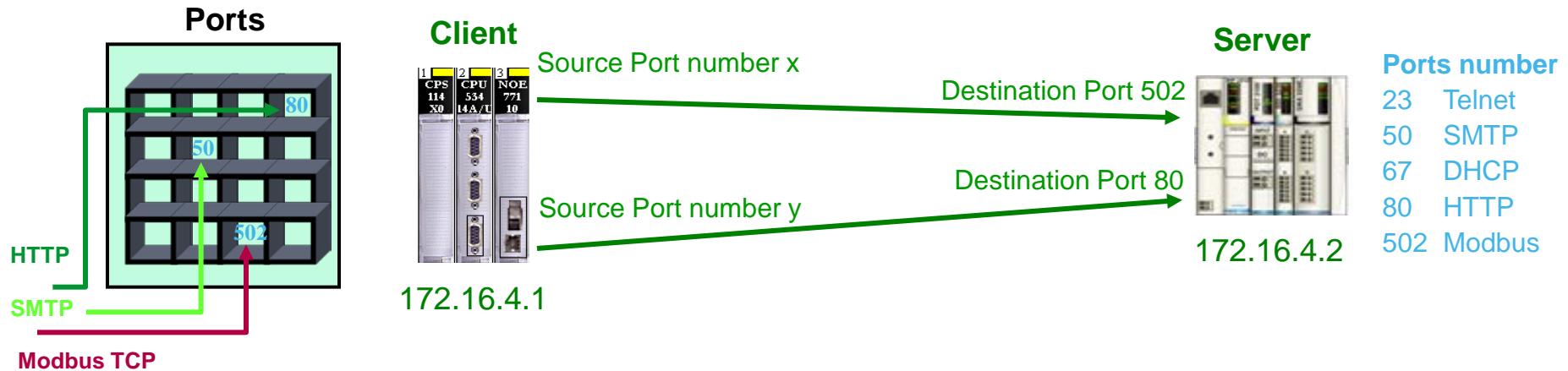
- Used in Different Levels



TCP (Transport Control Protocol)

● Port and socket concept

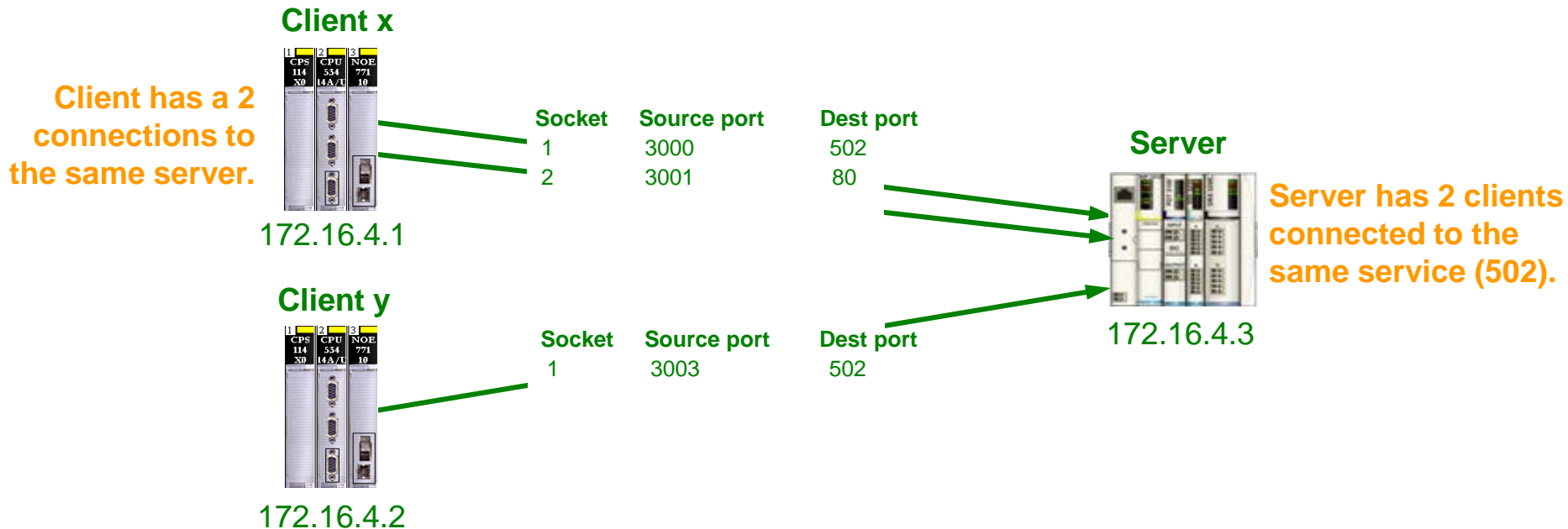
- Server devices may be running multiple services (Modbus, Web, Bootp, etc)
- These services listen on different TCP port for a request
- Client device sends an IP message which includes the source port number and destination port number to establish a connection between two devices (Socket connection).



TCP (Cont.)

- Socket management multiple connections

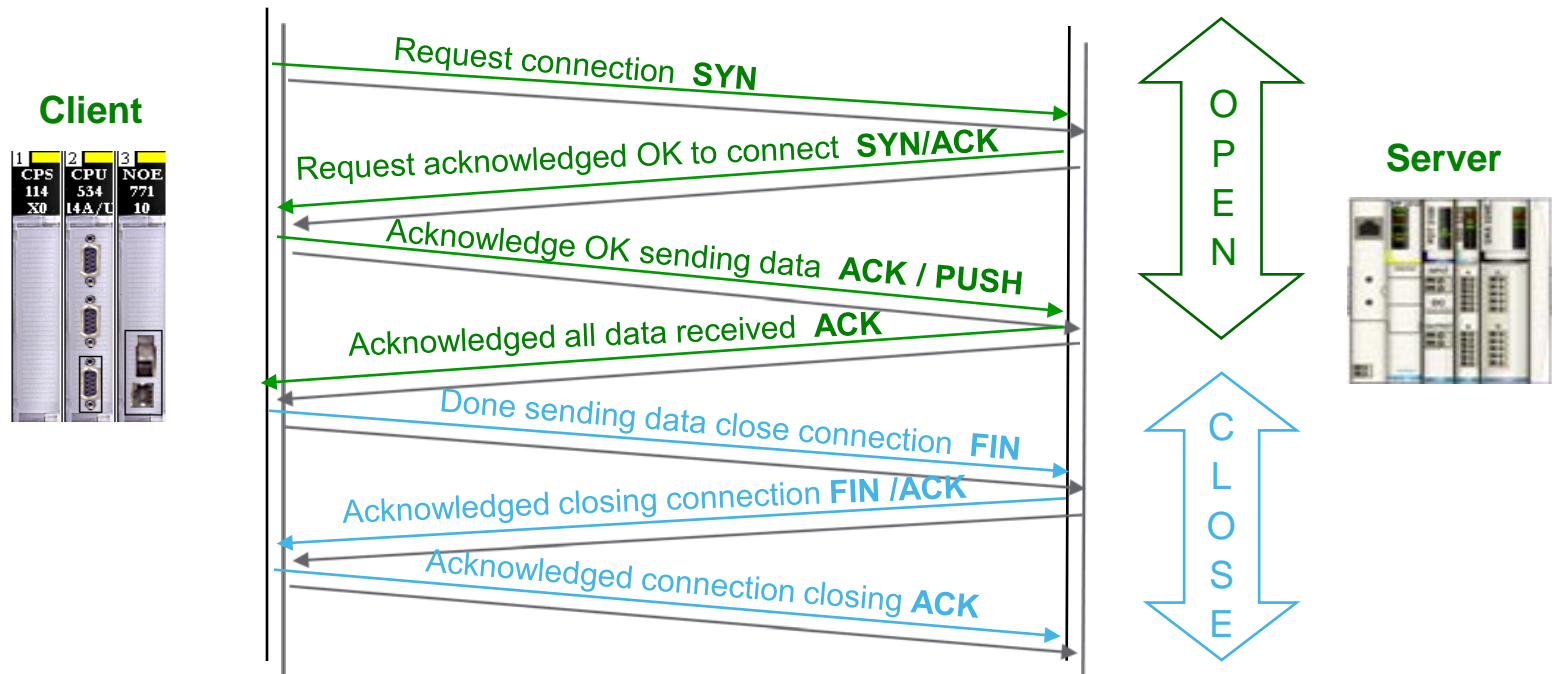
- A Client can have multiple connections to the same Server service
- A Server can have multiple Clients connected to the same service



TCP Connections

- Point to point messaging protocol

- Uses a handshake process to establish a connection
- Accounts for each byte of sent and received to guarantee delivery
- Connection is managed by setting bits in the TCP message to request (SYN), acknowledge (ACK), terminate or abort the connection (FIN)



IP (Internet Protocol)

- IP: Messaging Protocol Operating at OSI Layer 3 (Network layer)
- IP Characteristics
 - Responsible to transmit the message to the destination IP address to the local network or to the remote network.
 - Fragment a packet that is larger than the maximum transmission size unit (MTU) to send as multiple packets and de-fragment the packet at the destination level.
 - It makes no guarantee that the message will reach its destination. The reliability is done by the Transport layer

IP Address Assignment

- Default IP Address

- Derived from the factory and unique MAC address



Premium & M340 PLCs

MAC @ 00 - 80 - F4 - 12 - 0C - AF
IP @ = 85 . 16 . 12 . 175

- First 2 bytes are decimal **85 & 16**
- Hex to Decimal **conversion of last 2 bytes**

Quantum/Momentum/Advantys

MAC @ 00 - 00 - 54 - 12 - 0C - AF
IP @ = 84 . 18 . 12 . 175

- Hex to Decimal **conversion of last 4 bytes**

- Unique IP Addresses for Each Device

- In addition to MAC Hardware addresses, each devices is assigned a logical IP address.
- Assignment can be static or dynamic

Example : 139 . 160 . 12 . 85

The IP address contain 4 bytes and each byte has a range from 0 to 255.

Classful IP Address

- There are 4 main address classes
 - Class A - Used for larger networks
 - IP Address Range : 1.0.0.0 to 127.255.255.254
 - Default subnet mask : 255.0.0.0 - 16 777 216 hosts number
 - Class B - Used for Medium networks (ex. Schneider Electric)
 - IP Address range : 128.0.0.0 to 191.255.255.254
 - Default subnet mask : 255.255.0.0 – 65 534 hosts number
 - Class C - Used for smaller network and individual users
 - IP Address range : 192.0.0.0 to 223.255.255.254
 - Default subnet mask : 255.255.255.0 - 254 hosts number
 - Class D - Used for Multicast groups
 - IP Address range : 224.0.0.0 to 239.255.255.255
 - Devices cannot be assigned IP addresses in the multicast range
 - The Class E is reserved for the future : 240.0.0.0 to 255.255.255.255

Special IP Addresses

- Loopback

- IP address = 127.0.0.1
- Reserved for loopback (host connecting to itself)
 - Ex. A web browser connecting to a web server running on the same PC

- Broadcast

- IP address = 255.255.255.255
- Used by the network management or diagnostics is addressed to all devices on the network.

Subnet Mask

- Used to determine if the remote device is on a local or remote network
 - The mask separates the network portion of the IP address from the host portion of the IP address
 - The sending device uses its configured subnet mask to perform a Boolean **AND operation** with both its local IP address and the IP address of the remote device to connect to

Local device	IP Address	172.16.5.20		10101100.00010000.00000101.00010100
	Subnet Mask	255.255.255.0	AND	11111111.11111111.11111111.00000000
	Result			10101100.00010000.00000 101 .00000000

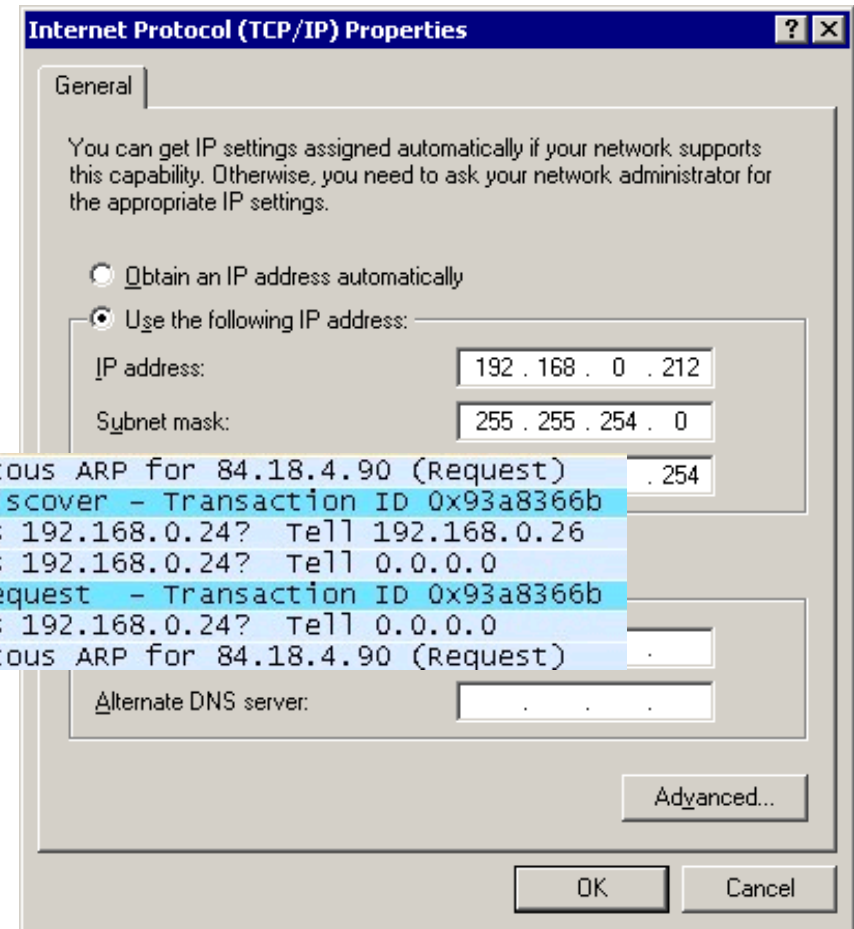
Remote device	IP Address	172.16.4.20		10101100.00010000.00000100.00010100
	Subnet Mask	255.255.255.0	AND	11111111.11111111.11111111.00000000
	Result			10101100.00010000.00000 100 .00000000

Troubleshoot TCP/IP

- In Windows

- Monitoring Free Tool: Wireshark

1	Modicon_12:04:5a	Broadcast	ARP	Gratuitous ARP for 84.18.4.90 (Request)
2	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x93a8366b
3	Telemeca_07:32:e5	Broadcast	ARP	who has 192.168.0.24? Tell 192.168.0.26
4	Modicon_12:04:5a	Broadcast	ARP	who has 192.168.0.24? Tell 0.0.0.0
5	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0x93a8366b
6	Modicon_12:04:5a	Broadcast	ARP	who has 192.168.0.24? Tell 0.0.0.0
7	Modicon_12:04:5a	Broadcast	ARP	Gratuitous ARP for 84.18.4.90 (Request)



 → <http://www.wireshark.org/>

Troubleshoot TCP/IP

● Ping Command

- Used to check a connection



- Launched from the DOS window (under XP)

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\sesa44505>ping 10.184.30.52

Pinging 10.184.30.52 with 32 bytes of data:

Reply from 10.184.30.52: bytes=32 time<1ms TTL=128
Reply from 10.184.30.52: bytes=32 time<1ms TTL=128
Reply from 10.184.30.52: bytes=32 time<1ms TTL=128
Reply from 10.184.30.52: bytes=32 time<1ms TTL=128

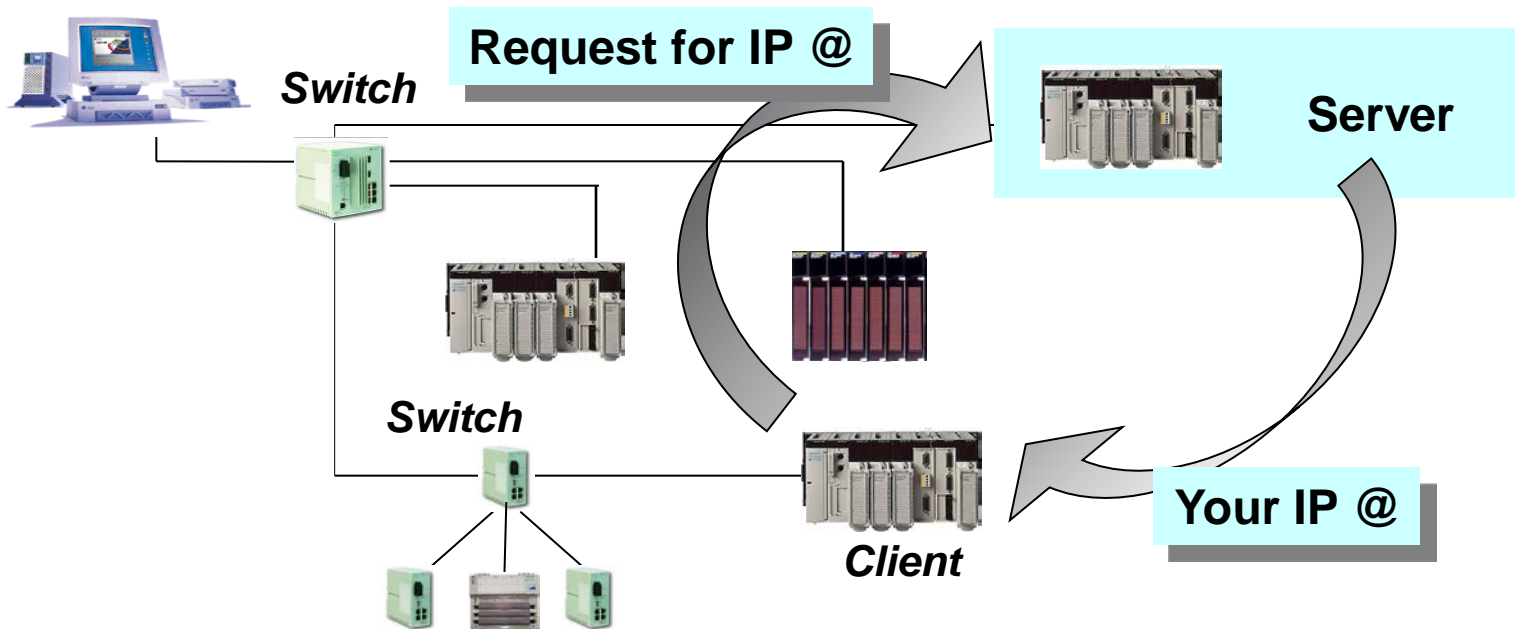
Ping statistics for 10.184.30.52:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```


Exercises

- IP Address (p4-19)
 - Find your IP address (record it for future use)
 - Change the IP address into 192.168.0.X1 or 192.168.0.1X1 (X=Group Number)
- Troubleshoot TCP/IP (p4-23)
 - Use PING command to ping your partner

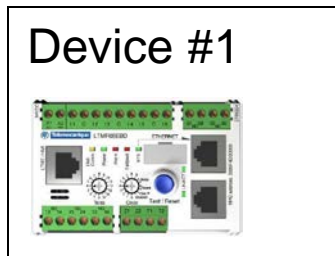
Dynamic Assignment

- IP Assignment Given By Servers



BootP Server

- Server Has a Populated List of Devices
 - Devices identified with their MAC address



My MAC address is 00.80.F4.FF.00.D5,
Can I get an IP Address?



Sure! You are listed in my address table.
Take this address: 192.168.0.23



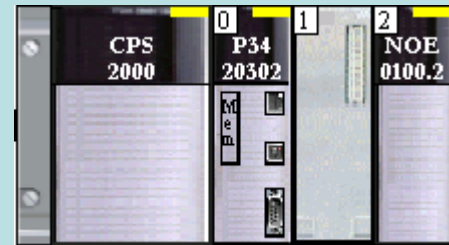
My MAC address is 45.80.F4.FF.33.12,
Can I get an IP Address?



Sorry, you are not listed into my address table.
Use your default IP address

Master (Server)

00.80.F4.FF.00.D5	192.168.0.23
00.80.F4.FF.44.21	192.168.0.81
00.80.F4.FF.F2.15	192.168.0.40



BootP Server (Cont.)

- Example in Unity Pro

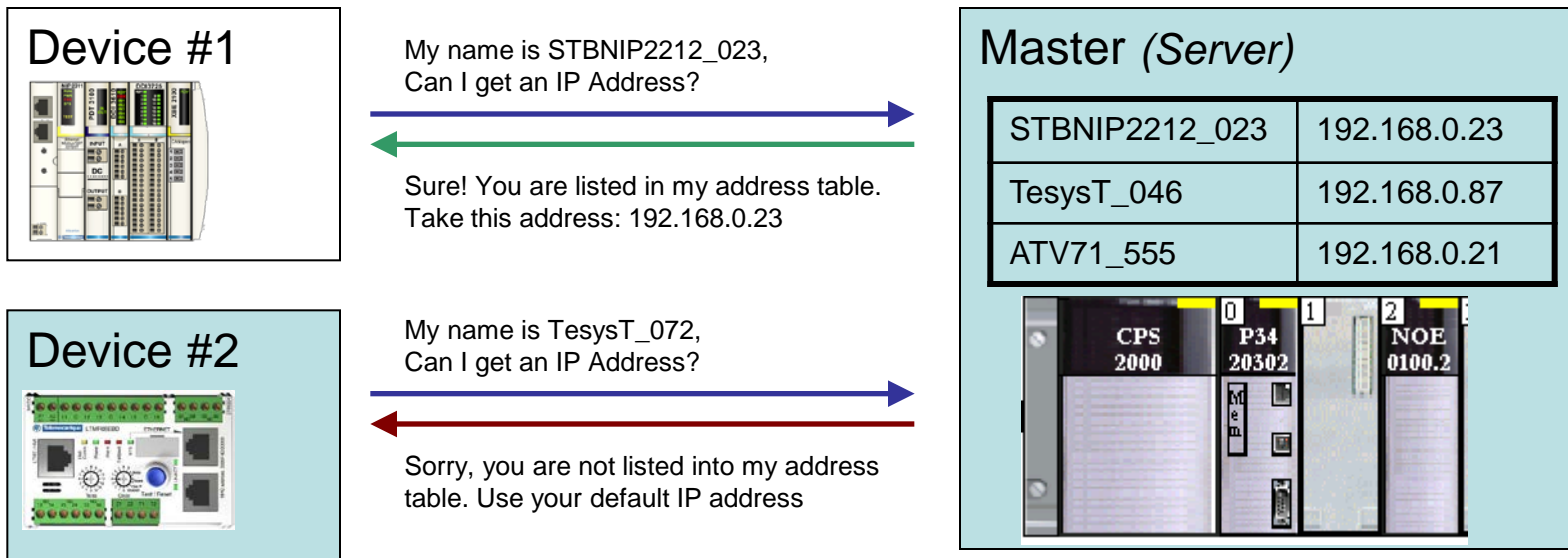
	Access	MAC address	Name	IP address	Netmask	Gateway
1	<input checked="" type="checkbox"/>	00.80.F4.FF.00.D5		192.168.0.23	255.255.255.0	0.0.0.0
2	<input checked="" type="checkbox"/>	00.80.F4.FF.44.21		192.168.0.81	255.255.255.0	0.0.0.0
3	<input checked="" type="checkbox"/>	00.80.F4.FF.F2.15		192.168.0.40	255.255.255.0	0.0.0.0
4	<input checked="" type="checkbox"/>					
5	<input checked="" type="checkbox"/>					

- Limitation: Device Cannot Be Changed without Reconfiguration

- As each device has a unique MAC address, replacing a failed device needs a reconfiguration of the BootP Address Server table.

DHCP Server

- Same As BootP but Based on Role Name
 - User configurable names used instead of MAC Addresses



DHCP: How To Create Role Names

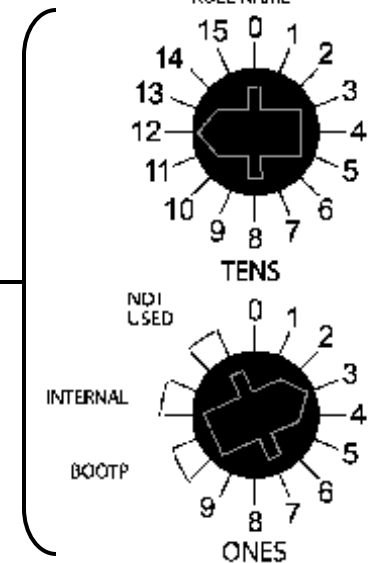
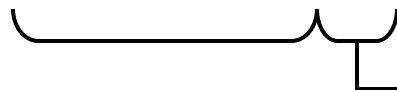
- Depends on The Device

- Using embedded HMI (ex: ATV71)
- Using web based applications (ex: Advantys STB)
- Using dedicated applications (ex: Unity Pro)
- Using hardware switches (ex: Tesys T, Advantys STB)

- Hardware Switches Example:

- Base name + Custom number derived from Switches

STBNIP2212_123



DHCP Server



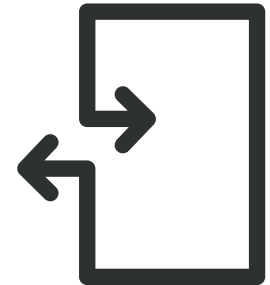
- Example with Unity Pro

	Access	MAC address	Name	IP address	Netmask	Gateway
1	<input checked="" type="checkbox"/>		STBNIP2212_123	192.168.0.23	255.255.255.0	0.0.0.0
2	<input checked="" type="checkbox"/>		TesysT_046	192.168.0.81	255.255.255.0	0.0.0.0
3	<input checked="" type="checkbox"/>		TCESGPA23F14F002	192.168.0.40	255.255.255.0	0.0.0.0

- Advantage: Easiness of Replacement

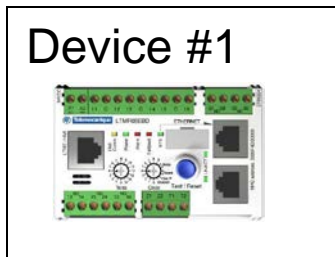
- Compared to the BootP method, replacing a faulty device with DHCP doesn't need to reconfigure the PLC application.

FDR Server



- Additional Feature of DHCP Server.

- Send the stored configuration inside a replaced device

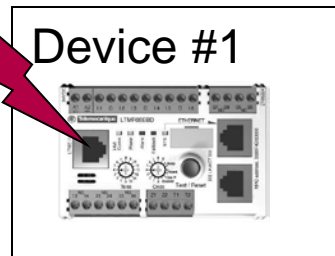


My name is TesysT_072,
Save my settings now!

Master (Server)

TesysT_072	192.168.0.87
------------	--------------

Saved Config. of TesysT_072

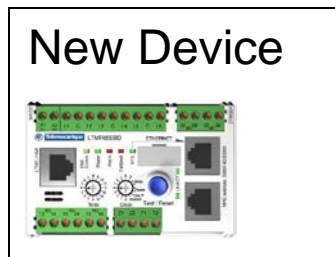


My name is TesysT_072,
Can I get an IP Address?

Master (Server)

TesysT_072	192.168.0.87
------------	--------------

Saved Config. of TesysT_072



Sure! You are listed in my table.
Take this address: 192.168.0.87

Moreover, I found a configuration
corresponding to your role name.
Here is your new configuration.

ARP

1	Modicon_12:04:5a	Broadcast	ARP	Gratuitous ARP for 84.18.4.90 (Request)
2	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x93a8366b
3	Telemeca_07:32:e5	Broadcast	ARP	who has 192.168.0.24? Tell 192.168.0.26
4	Modicon_12:04:5a	Broadcast	ARP	who has 192.168.0.24? Tell 0.0.0.0
5	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0x93a8366b
6	Modicon_12:04:5a	Broadcast	ARP	who has 192.168.0.24? Tell 0.0.0.0
7	Modicon_12:04:5a	Broadcast	ARP	Gratuitous ARP for 84.18.4.90 (Request)

● Address Resolution Protocol

- Request to obtain IP address
- Duplicate Address Check
 - Device issues ARP for the IP address it intends to take
 - If no response, the device assumes the IP address
 - If there is a response, the device should not assume the IP (duplicate address)

● Once the IP has been determined to be available

- Device issues a Gratuitous ARP (Includes Source IP and MAC address information)
 - Used to populate device list in the other devices and routers
- Advertises to others its availability on the network
 - Allows devices wishing to communicate with it that it is available

UDP (User Datagram Protocol)

- Transport protocol like TCP but without Acknowledgement
 - Provides an unreliable mechanism to transport data
 - Messages can be lost (not acknowledgment of the packet)
 - Retries and data integrity can be provided by the application layer
- Requires less processing overhead than TCP
 - Without ordering messages and managing connections. it is faster than TCP
 - The network interface does not have as much work to do as with TCP
- Allows the Broadcast message, unlike TCP that requires a dedicated socket between the end device. Applications example using UDP :
 - Simple Network Management Protocol (SNMP)
 - Network Time Protocol (NTP)
 - BootP and DHCP
 - Global Data (Real Time Publish Subscribe)

Network Design: Media

- Media

- Copper media
- Twisted pairs wires
- Easy to install : low cost installation



- Fiber optic

- Adapt different types of media
- Using over long distances or harsh industrial environment



Network Design: Copper Media

- Rated by category
- Categories established by ANSI/EIA/TIA Committee
 - Industrial applications recommend Shielded media for additional protection from external sources of interference
- Uses Standard 8 pin RJ-45 Connector
 - Cat 5 : Minimum required for 100Mbps Ethernet (unshielded cable)
 - Cat 5e : Enhanced for 100Mbps Full Duplex operation (shielded cable)
 - Cat 6 : New standard for 1 Gbs operation



The maximum length of one segment is 100 meters at 10, 100, 1000 Mbit/s.

Network Design: Fiber Optic

- Glass or plastic fiber designed to guide light along its length by total internal reflection.
- Permits digital data transmission over longer distances and at higher data rates than other forms of wired and wireless communications (up to 20km)



ST connector

Bayonet style connector



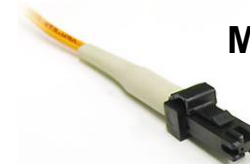
SC connector

Keyed square connector
May be coupled together



LC connector

Newer keyed Fiber connector
Smaller form factor : higher density



MTRJ connector

Small form factor, keyed connector

Network Design: Wifi

- **Wireless Communication**

- Available soon in Schneider Electric (already available with partners)
- Main issues are security and EMC behavior



Network Design: Components



Connexium Switch

- Hubs or switches

- Hubs are not recommended for industrial application



499NTR10100

- Transceivers

- Adapt different types of media
 - Ex. 499NTR10100 : converter electrical signal to optic signal



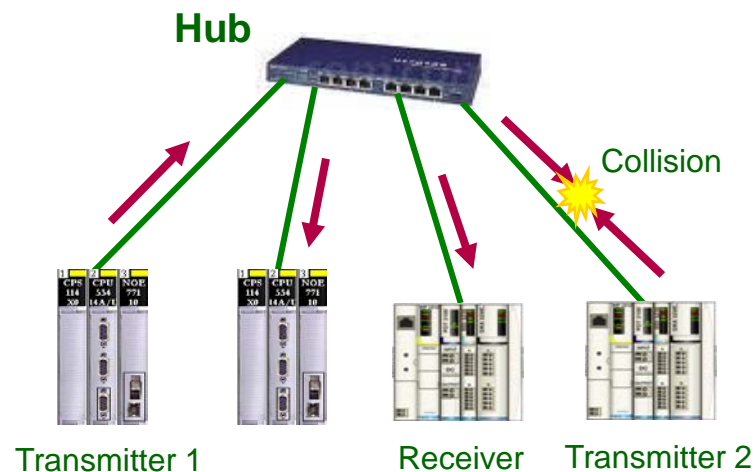
- Routers

- Used to route information between networks

Network Design: Hubs

- Half Duplex (Collisions)

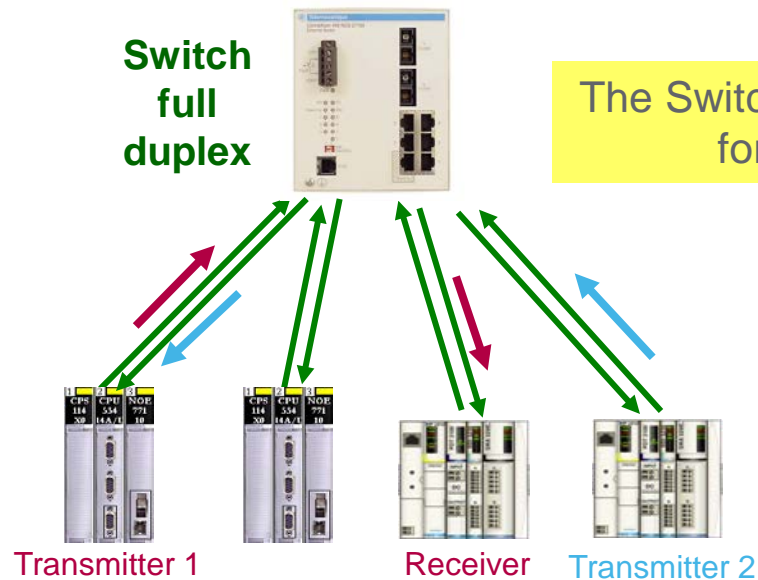
- Each and every 'frame' is repeated out all ports by hub
- All 'nodes' listen to see if they are the receiver of the frame
- As more nodes are added and collisions increase, performance decreases
- If two devices transmit simultaneously, a collision occurs



Network Design: Hubs

- Full Duplex (Collisions)

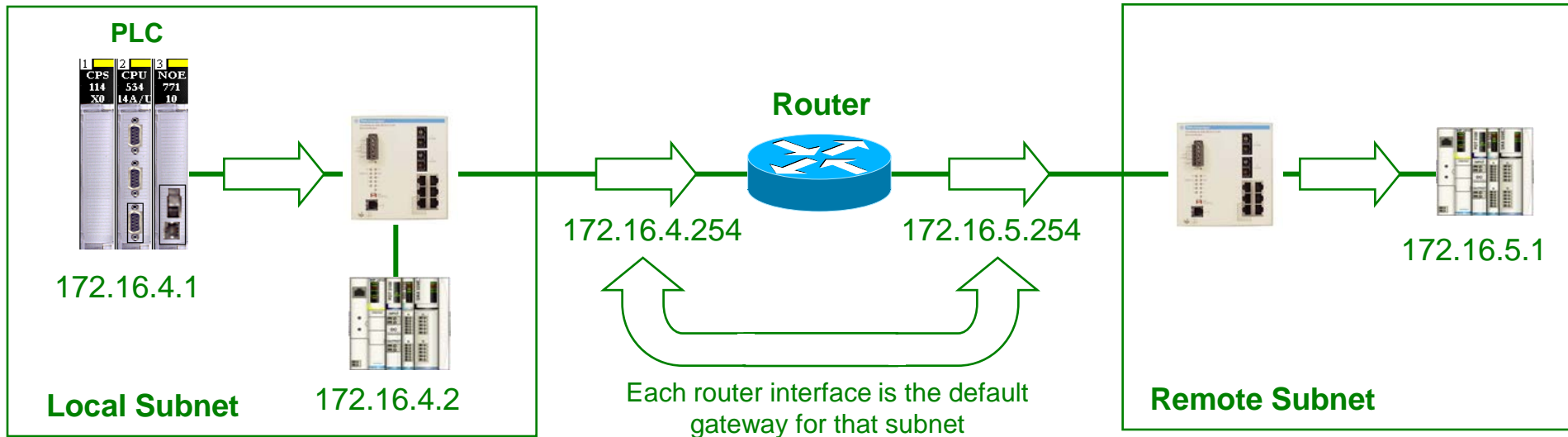
- Switch learn the addresses of the devices attached to them
- Switch full duplex can simultaneously transmit and receive a message



The Switches are extremely recommended for the Industrial application.

Network Design: Routers

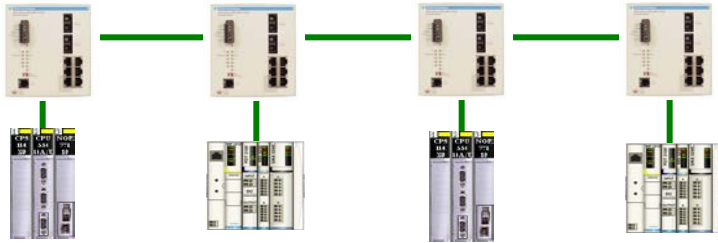
- Routing data between different sub networks
 - Act as a Gateway (IP address to be defined)
 - A router has an interface for each different networks. A table of those interfaces tells the router on which interface to send the packet



Nota : If there is not a router on the local network, the default gateway can be left blank or 0.0.0.0

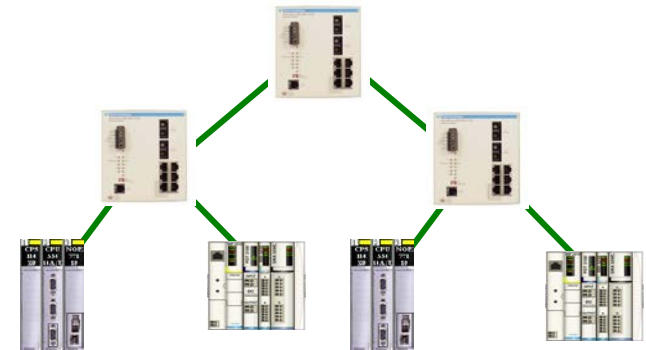
Network Design: Topologies

Bus topology



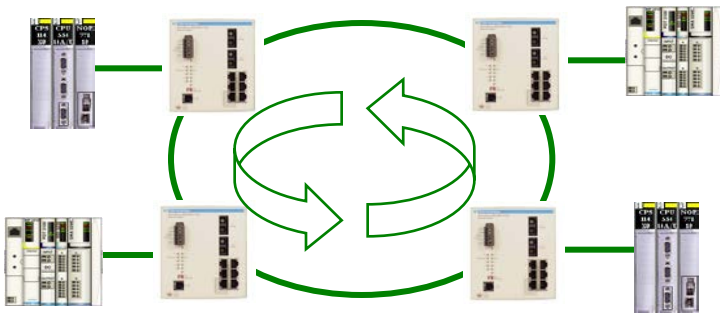
Switches in series, end devices connected to switches

Tree / Star topology



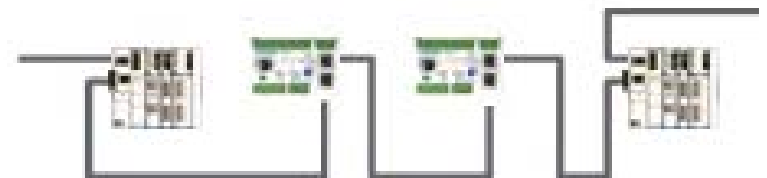
Switches in a star or tee-like arrangement

Ring topology



Switches form a ring or circle

Daisy Chain topology

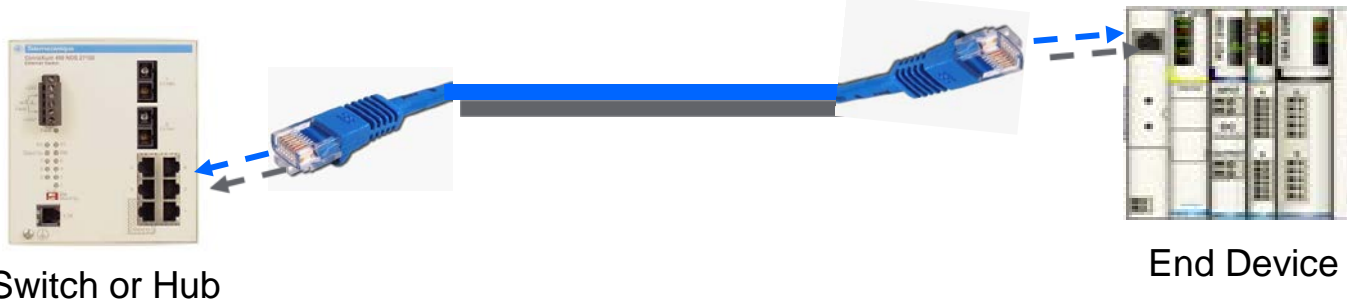


No switches needed

Network Design: Point to Point Cables

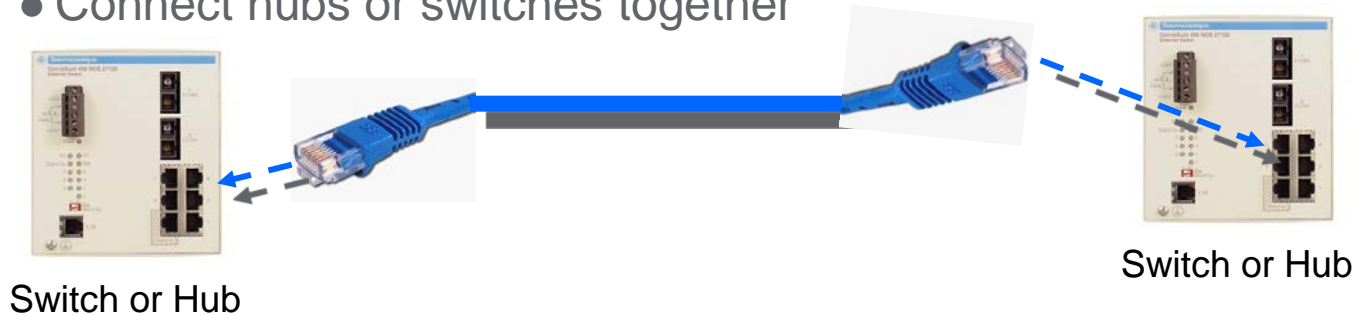
● Straight Cable

- Connect end devices to hubs or switches



● Crossed Cables

- Connect hubs or switches together

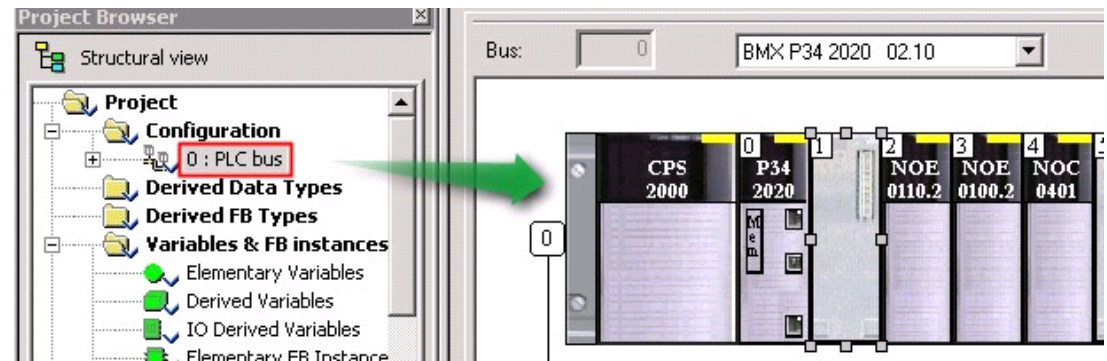


Programming with Unity Pro

2 Ways of Configuration

● Standard Ethernet Ports

- CPU Embedded Ports (BMXp342030, TSXP572634...)
- Ethernet Modules (BMXNOE0100, TSXETY5103...)
 - Configuration done from project browser

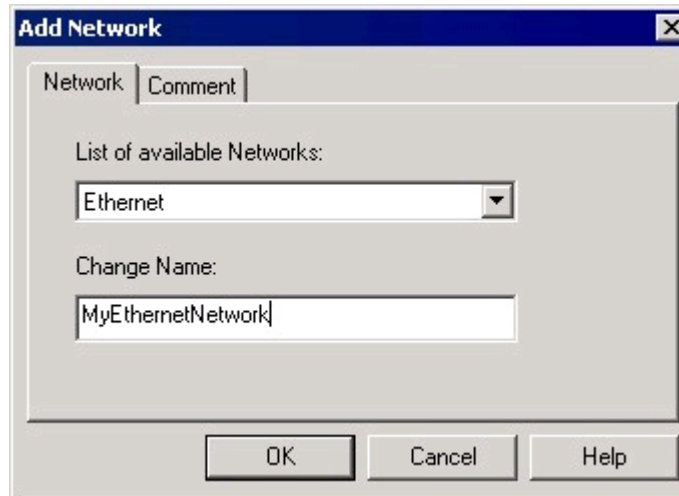


● DTM Compatible Ports

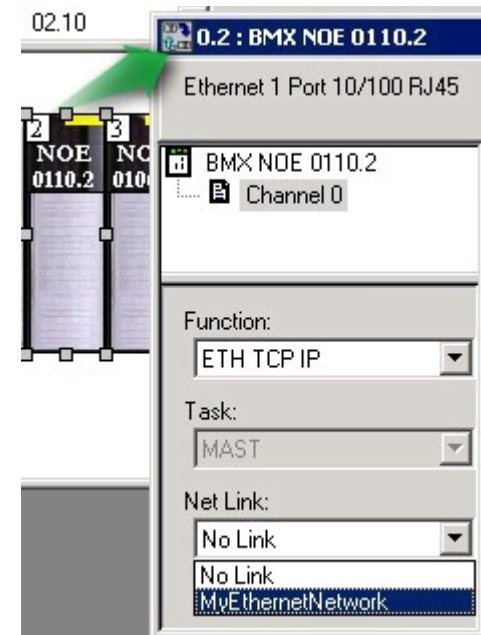
- Converged Modules (BMXNOC0401, TSXETC101, ...)
 - Configuration done from the DTM browser

Standard Configuration

- From Project Browser, Select the Network Component
 - Right click and create a new network



- Once Created, Link it to the Module



Network Configuration

- Options Depends on the Model

The image displays two screenshots of a network configuration interface, illustrating how options vary based on the selected model family.

Top Screenshot (Model Family: NOE 0100.2, NOE 0110.2):

- Model Family:** NOE 0100.2, NOE 0110.2
- Module Address:** Rack, Module, Channel
- Module IP Address:** IP Address (0 . 0 . 0 . 0), Subnetwork Mask (255 . 0 . 0 . 0), Gateway Address (0 . 0 . 0 . 0)
- Module Utilities:** NO (ID Scanning), NO (Global Data), NO (Address Server), NO (NTP)

Bottom Screenshot (Model Family: CPU 2020, CPU 2030 (>= V02.00), PRA 0100):

- Model Family:** CPU 2020, CPU 2030 (>= V02.00), PRA 0100
- Module Address:** Rack, Module, Channel
- Module IP Address:** IP Address (0 . 0 . 0 . 0), Subnetwork Mask (255 . 0 . 0 . 0), Gateway Address (0 . 0 . 0 . 0)
- Module Utilities:** NO (SMTP)

- Module's Utilities

Module's Utilities

- IP Configuration
- Messaging
 - Access rights from other devices.
- SNMP (Simple Network Management Protocol)
 - Used by software monitor and manage devices on an IP network. (ConnexView)
- SMTP (Simple Mail Transfer Protocol)
 - Parameters for sending and receiving e-mail.
- IO Scanning
- Global Data
- Address Server
 - configures the DHCP and BOOTP services included in the module.
- NTP (network time protocol)
 - Define the time synchronisation server
- Common Words (Premium Only)
 - used to configure words shared on EthWay
- Bandwidth
 - estimation of the Bandwidth taken by the different services.

IP Configuration

- Static (Configured)
- Dynamic (From a server)
 - Assign a role name (optional)

The screenshot displays a web-based configuration interface for IP settings. At the top, there are five tabs: 'IP Configuration' (selected), 'Messaging', 'IO Scanning', 'Global Data', and 'SNMP'. Below the tabs, the 'IP address configuration' section is visible. It contains two radio button options: 'Configured' (selected) and 'From a server'. Under the 'Configured' option, there are three input fields: 'IP address' with the value '192 . 168 . 0 . 55', 'Subnetwork mask' with the value '255 . 255 . 255 . 0', and 'Gateway address' with the value '0 . 0 . 0 . 0'. Below these fields is a 'Device Name' input field. The 'Ethernet configuration' section at the bottom shows two radio button options: 'Ethernet II' (selected) and '802.3'.

- Configuration Must Match with Rotary Switches!

Exercise

- Ethernet Configuration (p4-38)
 - Create a new project
 - Create a new network
 - M340 IP address = 192.168.0.X5
 - Test the connection
 - Configure the role name of the slave
 - Configure the DHCP: STB with IP address of 192.168.0.X2
 - Test the configuration

Explicit Exchange Data

● Function Blocks

- ADDM: Address Conversion : used to provide the target address for others function blocks (M)
- ADDR: Address Conversion : used to provide the target address (P)
- CREAD_REG: Continuous Register Reading (Q)
- CWRITE_REG: Continuous Register Writing (Q)
- DATA_EXCH: Exchanging Data (M,P)
- MBP_MSTR: Multiple Functions – (Q)
- READ_REG: Read Register (Q)
- READ_VAR: Reading variables (M, P)
- SEND_EMAIL: Sending Email (M)
- SYMAX_IP_ADDR: target address to reach legacy SquareD PLC (Q)
- TCP_IP_ADDR: used to provide the target address (Q)
- WRITE_REG: Write Register (Q)
- WRITE_VAR: Writing variables (M, P)



ADDM FB

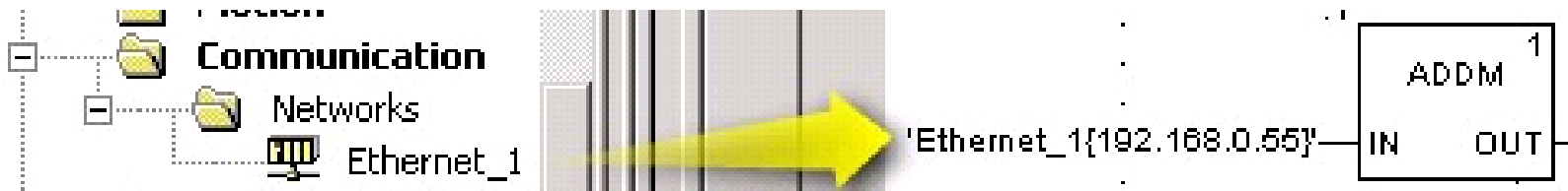


- IN:

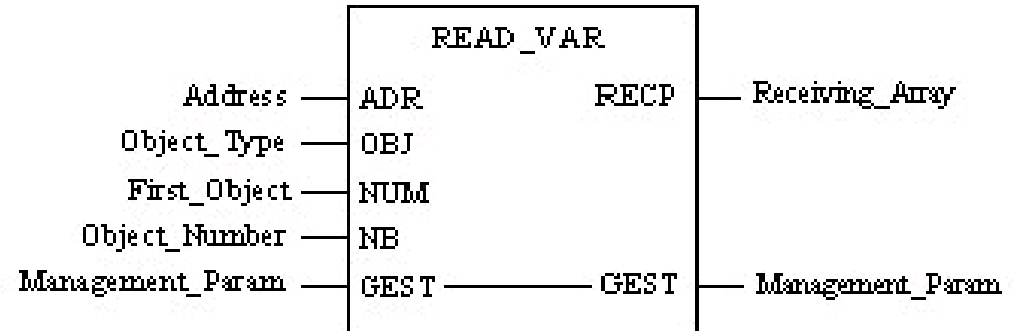
- String variable: 'NameOfTheNetwork{TargetIPAddress}'

- Out

- Converted table of words, to be used in the others FB



READ_VAR FB



● IN:

- **ADR:** to be linked to the output of the ADDM block.
- **OBJ:** defining object to read (in the case of Modbus register: '%MW')
- **NUM:** starting register to read
- **NB:** number of consecutive register to read

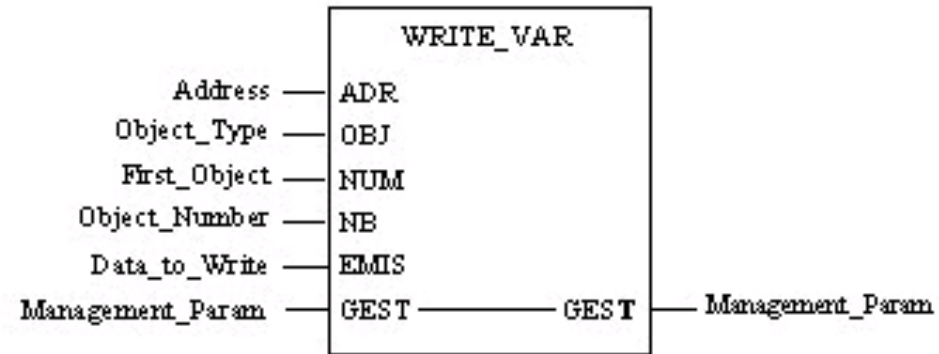
● OUT

- **RECP:** reception zone of the block, delivering the value read (table of words)

● IN/OUT

- **GEST:** table of 4 words to manage the communication block (errors, timeout, length, etc..)

WRITE_VAR FB



● IN:

- **ADR:** to be linked to the output of the ADDM block.
- **OBJ:** defining object to read (in the case of Modbus register: '%MW')
- **NUM:** starting register to read
- **EMIS:** source table to write from the PLC

● IN/OUT

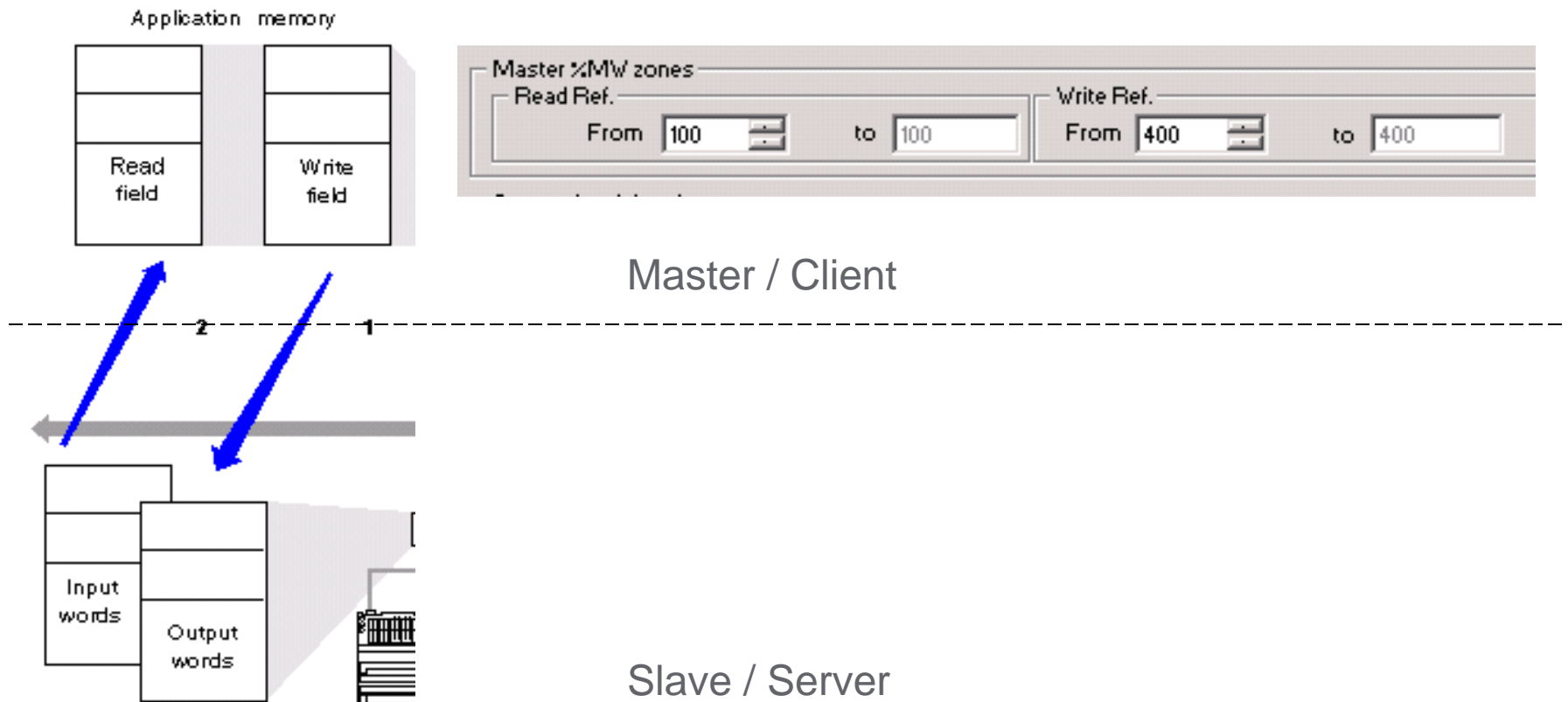
- **GEST:** table of 4 words to manage the communication block (errors, timeout, length, etc..)

Exercise

- Sending Requests (p4-42)
 - Enable “Allow Dynamic Arrays” in Project Settings
 - Insert the ADDM FB
 - Insert READ_VAR and WRITE_VAR FB
 - Test the configuration

Implicit Exchange: IO Scanning

- Table of Words Exchanged Automatically Between Master & Slave



IO Scanning

- Filling the Different Fields

IP Configuration | Messaging | **IO Scanning** | Global Data | SNMP | Address Server | ICP | Bandwidth

Master 32-bit zones

Read list: From 100 to 243 Write list: From 400 to 114 Repetitive rate step: 3

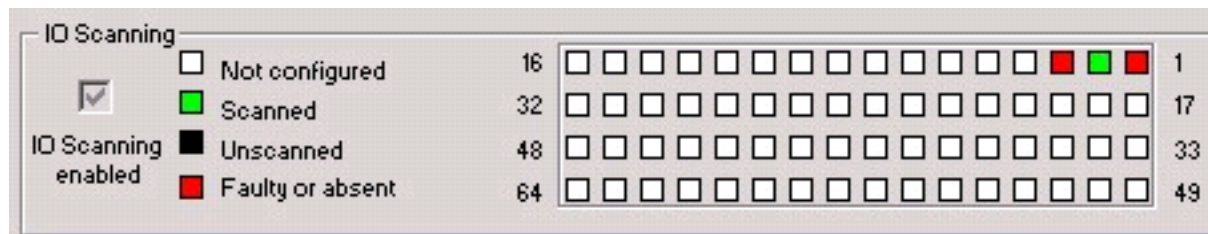
Scanned peripherals

	IP address	Device Name	Unit ID	Slave Station	Health Timeout (ms)	Repetitive rate (ms)	FD Master Object	FD Ref Slave	FD length	Last value (input)	WR Master Object	WR Ref Slave	WR length	Desc
1	192.168.0.23	PRM_Master	255	Index	1500	25	32M/100	0	40	Set to 0	32M/100	4096	50	
2	192.168.0.43		255	Index	1500	63	32M/160	1012	4	Hold last	32M/160	34	65	
3	192.168.0.62	STAMP	255	Index	1500	63	32M/164	5381	100	Hold last	32M/164	0	100	
4														
5														

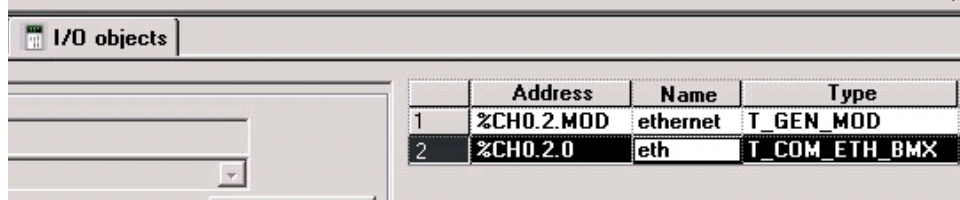
1 2 3 4 5 6 7 8 9 10 11 12 13 14

IO Scanning (Cont.)

- Debug View Online Mode (M340 and Premium Only)



- IODDT Can Be Configured



Name	Value	Type	Comment
ethernet		T_GEN_MOD	
MOD_ERROR	0	BOOL	Module error
EXCH_STS	0	INT	Exchange status
STS_IN_PROGR	0	BOOL	Status parameter read in progress
EXCH_RPT	0	INT	Channel report
STS_ERR	0	BOOL	Error while reading module status
MOD_FLT	0	INT	Module faults
MOD_FAIL	0	BOOL	Internal fault: Module failure
CH_FLT	0	BOOL	Faulty channel(s)
BLK	0	BOOL	External fault: Terminal block
CONF_FLT	0	BOOL	Hardware or software configuration fault
NO_MOD	0	BOOL	Module absent or powered-down
EXT_MOD_FLT	0	BOOL	FIPIO extension module fault
MOD_FAIL_EXT	0	BOOL	Internal fault: Module failure (only FIPIO extension)
CH_FLT_EXT	0	BOOL	Faulty channel(s) (only FIPIO extension)
BLK_EXT	0	BOOL	External fault: Terminal block (only FIPIO extension)
CONF_FLT_EXT	0	BOOL	Hardware or software configuration fault (only FIPIO extension)
NO_MOD_EXT	0	BOOL	Module absent or powered-down (only FIPIO extension)
eth		T_COM_ETH_B...	
CH_ERROR	0	BOOL	Channel error
SERVICES_STS	-20	INT	Status of the different services
P502_STATUS_BIT	0	BOOL	Port 502 messaging service status (0=OK, 1=NOK)
IOS_STATUS_BIT	0	BOOL	IO Scanner service status (0=OK, 1=NOK)
IGLBD_STATUS_BIT	1	BOOL	Global Data service status (0=OK, 1=NOK)
EMAIL_STATUS_BIT	1	BOOL	Email service status (0=OK, 1=NOK)
FDRS_STATUS_BIT	0	BOOL	FDR Server service status (0=OK, 1=NOK)

IO Scanning Specific Integration Tool

- Only with Compatible Devices
 - PRM / OTB / STB

	IP address	Device Name	Unit ID	Slave Syntax	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	F
1	192.168.0.22	...	255	Index	1500	60	%Mw/200	535
2		...						
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								

Property	
Device Type	Device Name
...	
STB	
OTB	
DTM	
Data Exchange Required	
Input words	0
Output words	0
OK	Launch No Tool
Update	Close

Exercise

- IO Scanning Configuration (p4-48)
 - Configure IO Scanning
 - Testing IO Scanning
 - Monitoring IO Scanning Status
 - Retrieving Process Data Values

- Advantys STB Integration Tool (p4-52)
 - Create the device
 - Test the configuration

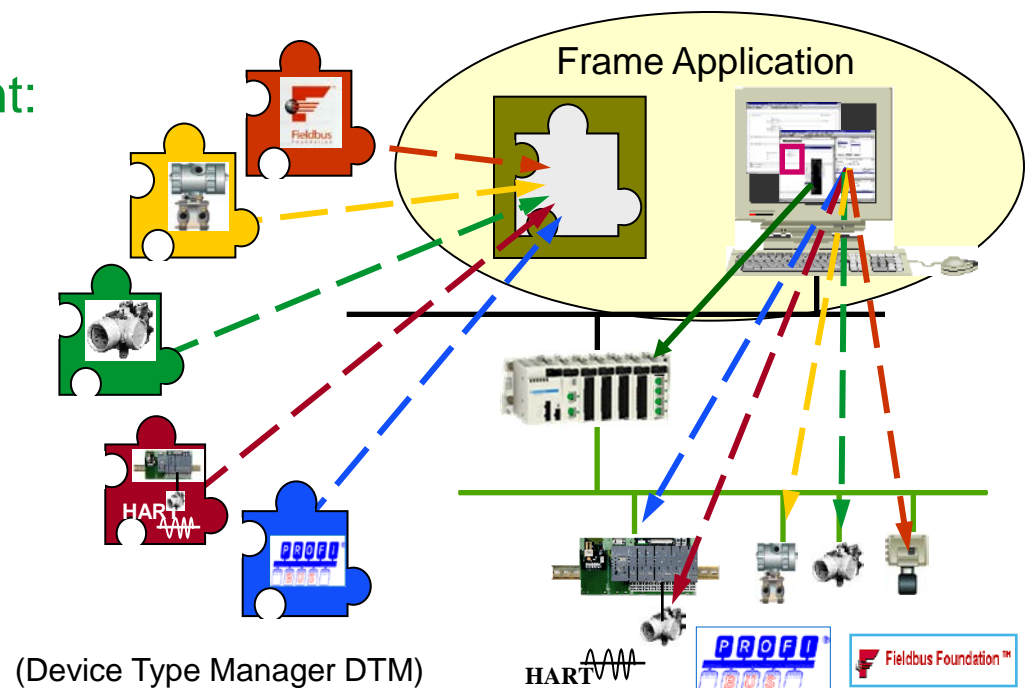
FDT / DTM Configuration

- **Field Device Tool**

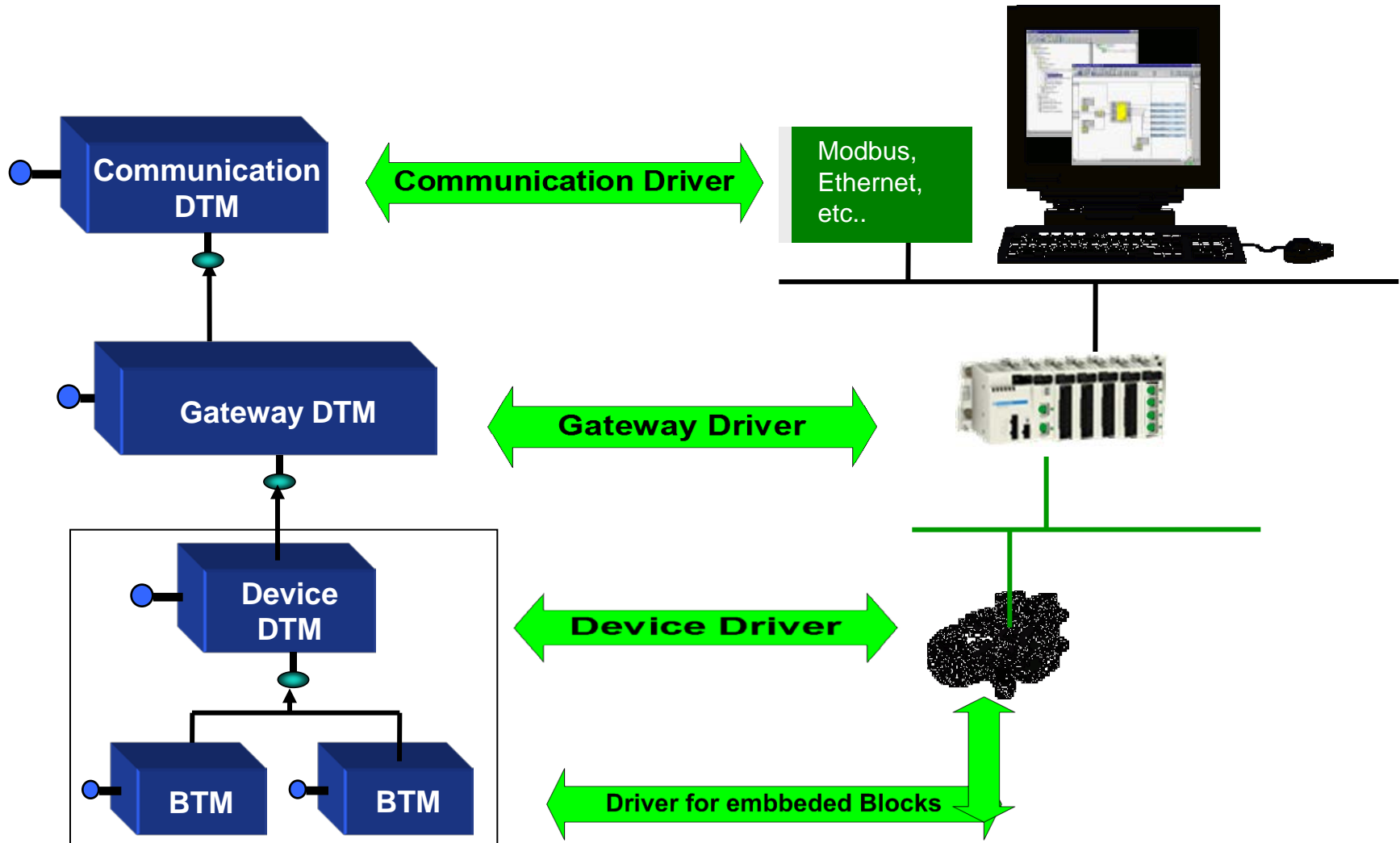
- Vendor tools integrated inside one tool
- Tools are Device Type Managers (DTM)

- **Benefit: Common Environment:**

- User Management
- DTM Management
- Data Management
- Network Configuration
- Navigation



Device Tool Managers

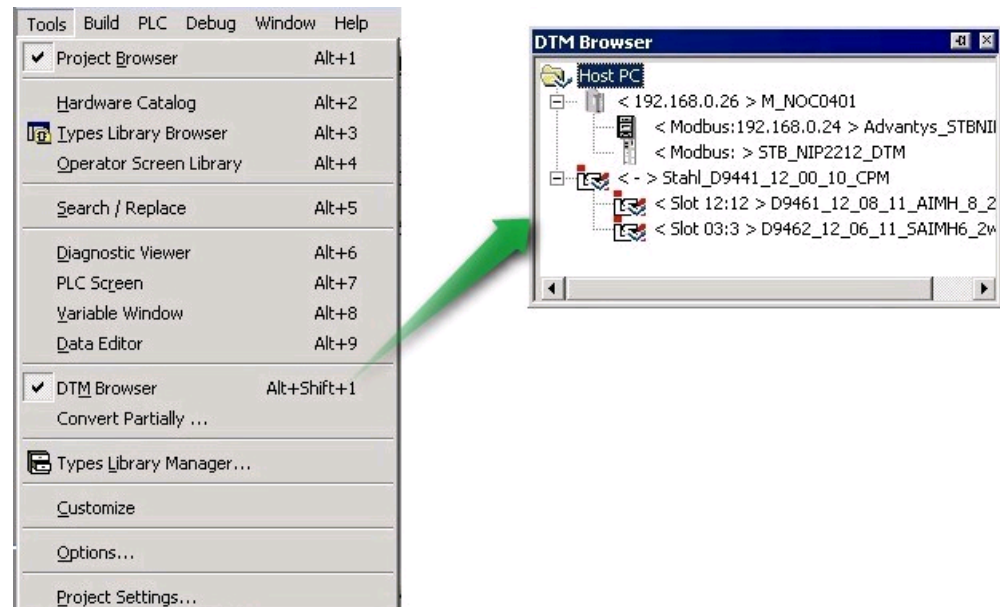


DTM in Unity Pro

● DTM Browser

- Integrated DTM (implemented when module added from the PLC_Bus)
- Dedicated DTM (devices that can interact with Unity Pro application)
- Third Party DTM (Unity Pro used as any other FDT frame application)

● Tools – DTM Browser



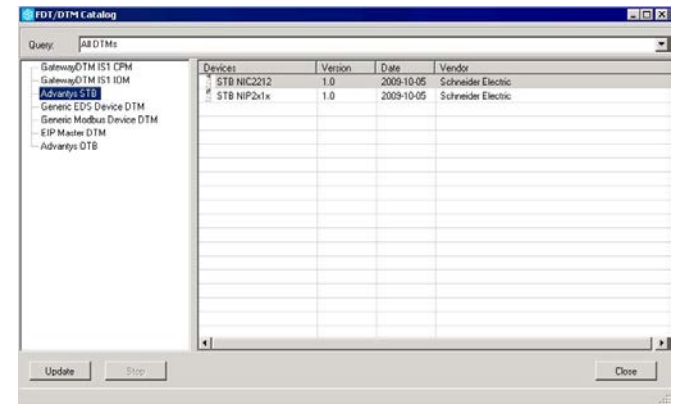
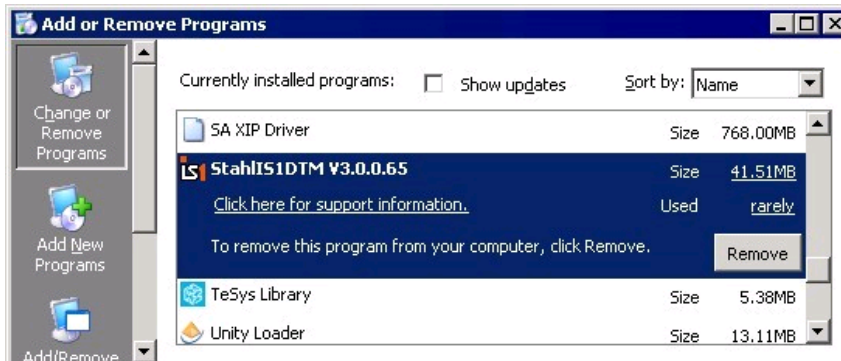
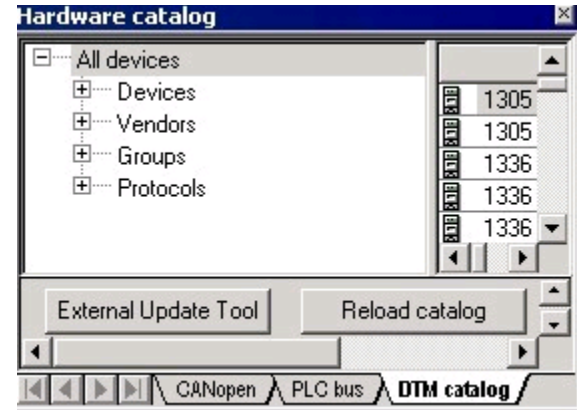
Hardware Catalog

- DTM Database

- Tools Menu – Hardware Catalog

- Update Tool

- Used to add New devices inside the catalog
- DTM must be installed in Windows before



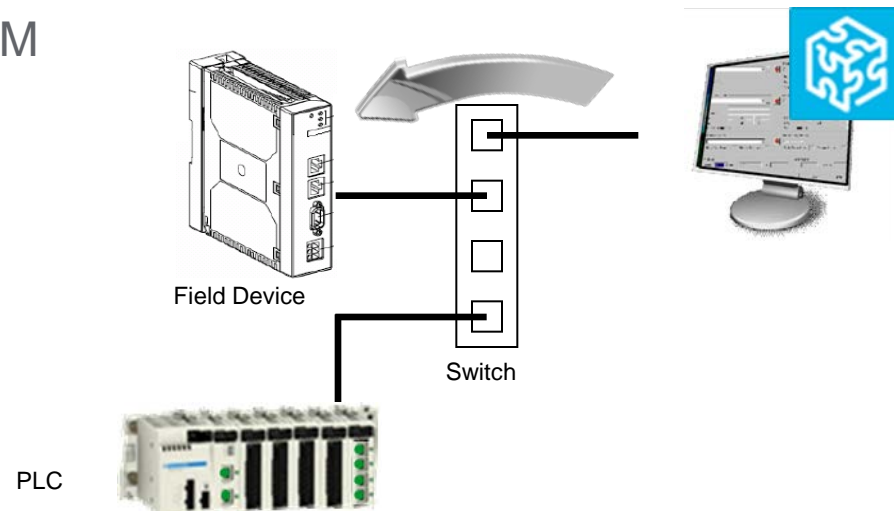
- DTM Config saved into STU or STA (not XEF)
- DTM must be installed on the computer (otherwise project cannot be saved)

DTM Transfer

- With Project Transfer
 - Integrated DTM

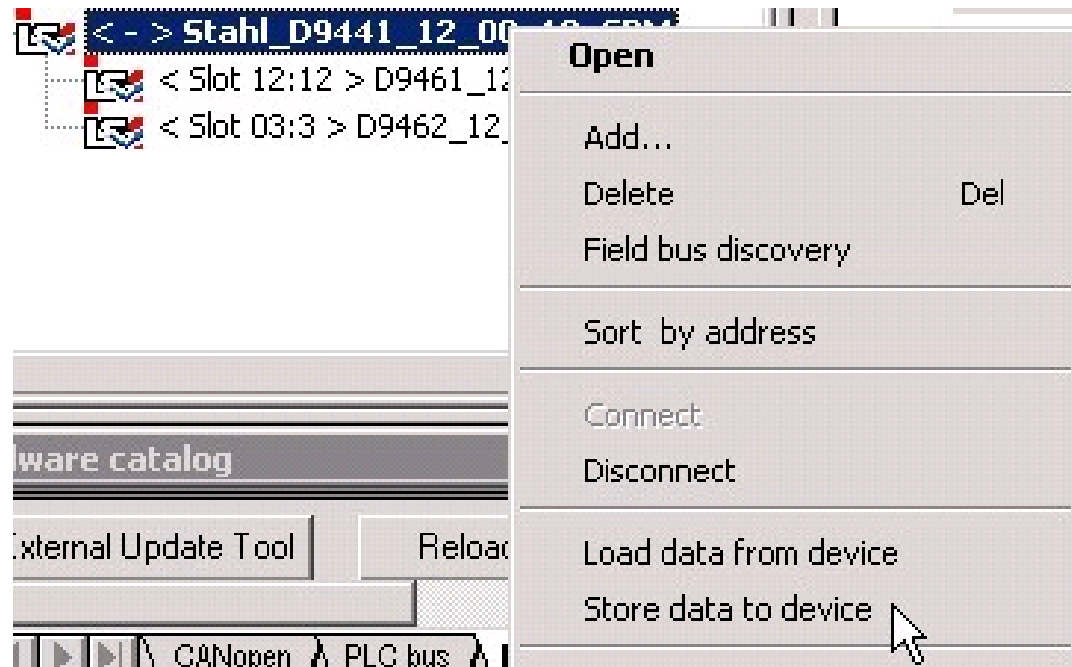


- Direct Ethernet Communication
 - Dedicated or Third Party DTM



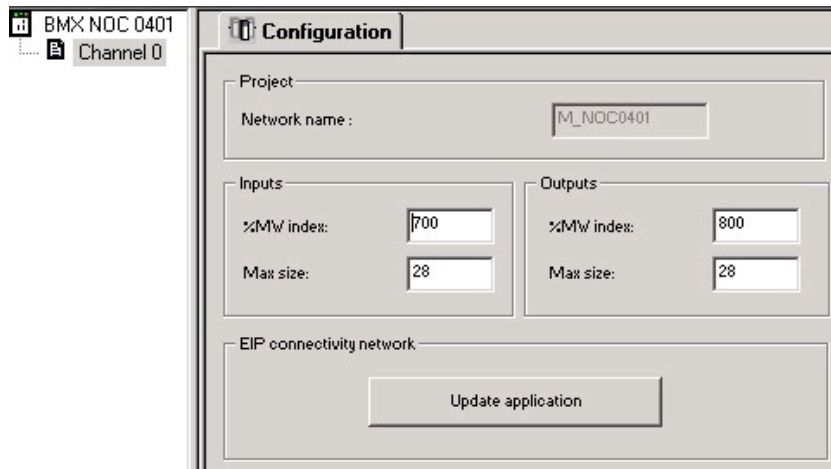
DTM Transfer for Non Integrated DTM

- From DTM Browser, Select Connect then Store data to device



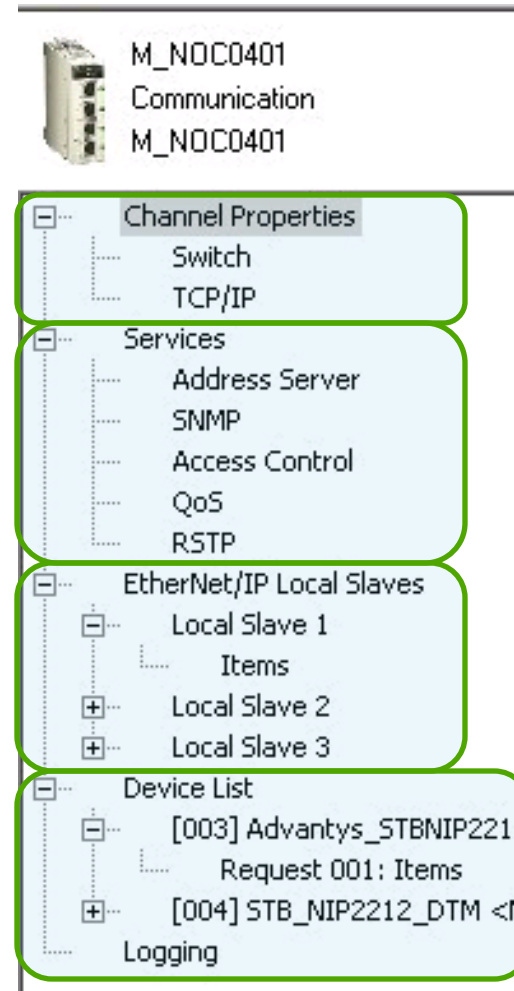
NOC0401 Configuration

- Added in Unity Pro



PLC_Bus Viewer

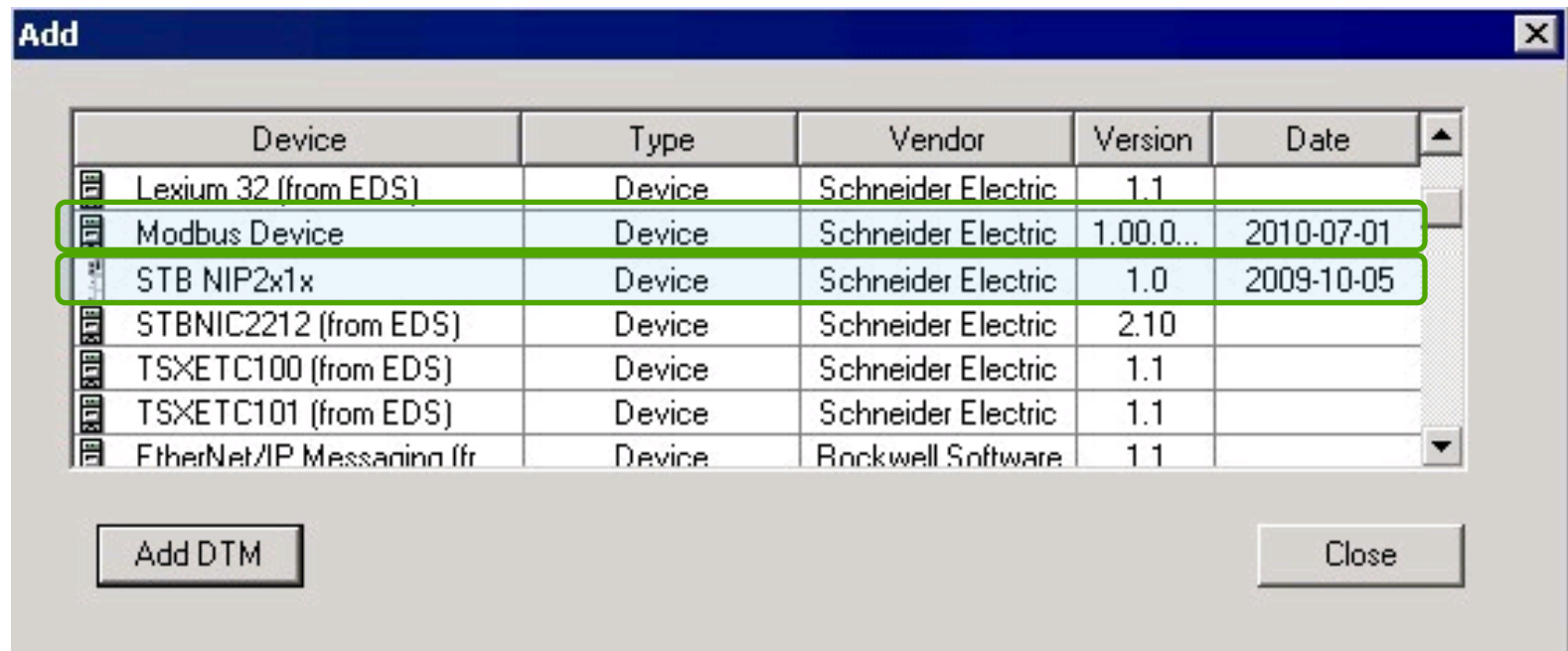
- Integrated DTM Module



DTM Browser

Add a Modbus Slave

- Generic Modbus Slave DTM
- Installed Modbus Slave DTM
 - Example: Advantys STB DTM



Configure the Slave

- Properties

- Customize IODDT Variable Names

- Address Setting

- Enter static or dynamic IP configuration

- Request Setting

- IO Scanning configuration inside the module

The screenshot shows the Unity Pro software interface with the 'Request Setting' tab selected. The left sidebar shows a tree view with 'Device List' expanded, listing several devices. The main area displays a table with the following data:

Connection Bit	Unit ID	Health Time Out(ms)	Repetitive Rate(ms)	RD Address	RD Length	Last Value	WR Address	
1	255	1500	60	8603	1	Hold Value	8601	

Monitoring Connection

● Connection Bit

- Monitoring if device is present

Name	Value	Type
managePara		ARRAY[0..3] OF...
Received_Data		ARRAY[0..49] O...
%m22	0	EBOOL
M_NOC0401_7_IN		T_M_NOC0401...
HEALTH_BITS_IN		ARRAY[0..31] O...
HEALTH_BITS_IN[0]	2#0111_0011	BYTE
HEALTH_BITS_IN[1]	0	BYTE

● Control Bit

- Enable / Disable IO Process Data update

Name	Value	Type
%m22	0	EBOOL
M_NOC0401_7_IN		T_M_NOC0401...
M_NOC0401_7_OUT		T_M_NOC0401...
CONTROL_BITS_OUT		ARRAY[0..31] O...
CONTROL_BITS_OUT[0]	0	BYTE
CONTROL_BITS_OUT[1]	0	BYTE
CONTROL_BITS_OUT[2]	0	BYTE
CONTROL_BITS_OUT[3]	0	BYTE
CONTROL_BITS_OUT[4]	0	BYTE
CONTROL_BITS_OUT[5]	0	BYTE

Exercises

- Change the Device Role Name (p4-64)
 - Assign new name to distinguish from previous exercise
- Installing the DTM (p4-65)
 - Installing DTM & Update the hardware catalog
- Configuring NOC0401 and Modbus TCP/IP (p4-67)
 - Insert BMXNOC0401 and configure it
 - Add a Generic Modbus device and configure it
 - Create custom variable names
 - Test the application
 - Do the same by configuring device from DTM

EtherNet Modbus TCP

- Main Features Reminder:
 - up to 100m in 100BASE TX (RJ45 cables) at 100Mbit/s - up to 1Gbit/s
 - Performances <10ms if 802.1Q is respected (with Connexium switches)
 - Thousands of devices (depends on the IP config.) but IO Scanning Limitations
 - Max 240 bytes per telegram
 - No separate ground line
 - Excellent error detection thanks to TCP
 - Configuration by DTM
 - Easy to debug (no need of additional hardware on the computer)