# Profinet IO-Device Emulator based on the Man-in-the-middle Attack

Michel Baud, Max Felser
Berne University of Applied Sciences
Engineering and Information Technology
Division of Electrical and Communication Engineering
Jlcoweg 1, CH-3400 Burgdorf, Switzerland
michel.baud@bfh.ch, max.felser@bfh.ch

## Abstract

*A Profinet IO Network of Class A uses standard Ethernet network components. If these network components can be attacked by e.g. a man-in-the-middle attack, this is also the case for Profinet IO. This paper outlines these possibilities and shows, how this may be used to build IO-Device emulators for system-testing.*

## 1. Introduction

An Ethernet Network can be hacked by using the man-in-the-middle (MITM) attack. This can occur when one device in the network, typically a PC, presents itself on the network to the two stations as being the other station. Thus, the two devices have the impression that they are communicating with each other, but in fact they are communicating only with the man-in-the-middle PC.

This type of attack is well known in the IT-World as "ARP poison" or "Port stealing" [4]. But in the world of automation technology, fieldbuses like Profibus [5] are used for industrial networks. This type of attack is currently unknown for fieldbuses.

Ethernet based fieldbuses have been developed over the past years and are now being considered for application in the field of automation technology. This exposes real-time automation and control installations as a new application of these attacks.

For the usage of Ethernet technology in Real-Time Applications there are different propositions to modify the standard Ethernet protocol in order to accomplish Real-Time Ethernet (RTE) [6]. But is it also possible to attack these RTEs with the same methods known to the IT world? This would raise new questions about security and up-time of Ethernet based Fieldbuses.

Profinet IO RTE is regarded as one of the leading solutions for the European market. A Profinet IO network of class A uses standard Ethernet switches for the setup of an automation network. A Profinet IO system is built with an IO-Controller which controls one or several IO-Devices over Ethernet connections [1] (see Figure 1).

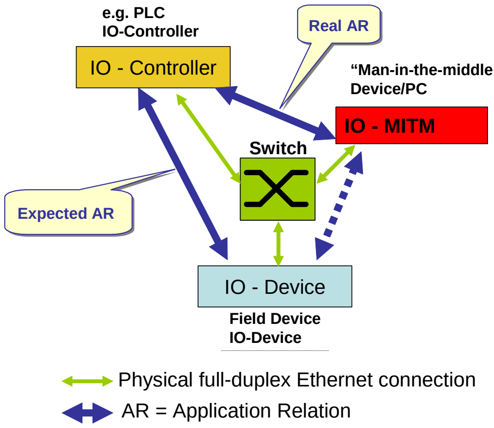

**Figure 1: PROFINET IO Systemlayout**

We will now summarise how a Profinet IO system communicates, and secondly, we will demonstrate if this attack is applicable to Profinet IO systems. Finally, we will show possible approaches to use this attack method to build system test devices.

## 2. Profinet IO communication setup

A Profinet IO system consists of an IO-Controller and one or several IO-Devices and one or several IO-Supervisors. The IO-Supervisors are typically engineering tools. For this study we do not further consider these IO-Supervisors and we focus on the connection of the IO-Controller to the IO-Device(s). This connection is called an Application Relation (AR).

### 2.1. Name assignment

The IO-Controller sets up an AR to every IO-Device it intends to control. For this a UDP/IP transmission with Distributed Computing Environment / Remote Procedure Calls (DCE RPC) is used. The possible IP addresses are defined in the planning phase of the IO-Controller with the Engineering Tool. The IO-

Controller has the task to assign the IP addresses to the different IO-Devices.

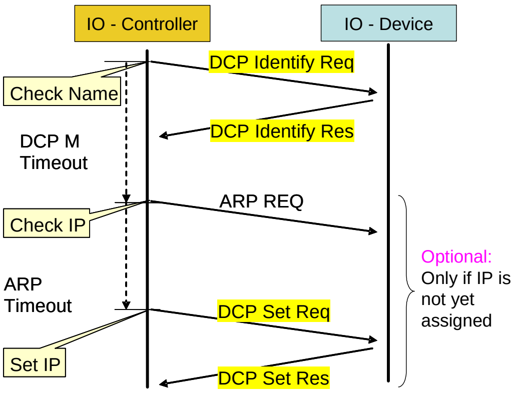

**Figure 2: Assignment of the IP address to the IO-Device, normal sequence**

The identification of the IO-Devices is not performed by the use of IP addresses, but rather with their Profinet names. During the engineering phase the Engineering tool assigns a name to every IO-Device using the DCP (Discovery and Basic Configuration) protocol (see Figure 2). This name is stored in a non volatile memory in the IO-Device and must be unique for the whole network.

### 2.2. AR Setup

When the IO-Controller starts processing, it first verifies that all IO-Devices are reachable with their names using the DCP-Identify. If an IO-Device has not yet had its IP address assigned, the IO-Controller verifies with an ARP-Request, that the name is not already in use and then assigns a valid IP address to the IO-Device using a DCP-Set. The IO-Controller then sends the AR-Setup data over the assigned IP address. Inside the AR the IO-Controller specifies a Communication relation (CR) for the cyclic IO-data to be exchanged between the IO-Controller and the IO-Device.

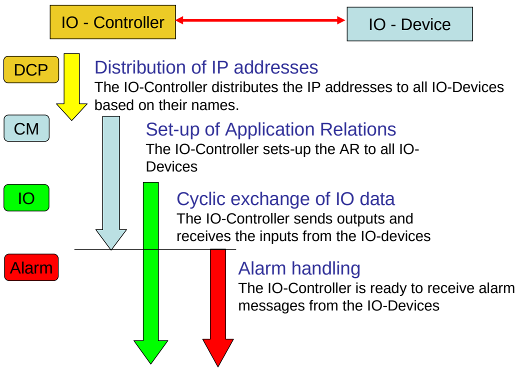
IO - Controller
IO - Device
DCP
Distribution of IP addresses
The IO-Controller distributes the IP addresses to all IO-Devices based on their names.
CM
Set-up of Application Relations
The IO-Controller sets-up the AR to all IO-Devices
IO
Cyclic exchange of IO data
The IO-Controller sends outputs and receives the inputs from the IO-devices
Alarm
Alarm handling
The IO-Controller is ready to receive alarm messages from the IO-Devices

**Figure 3: AR Setup Sequence**

Once the initialization sequence is complete (refer to Figure 3) the IO-Device sends its input data in the specified cycle time to the IO-Controller and the IO-Controller sends the output data to the IO-Device over a Profinet RT-Frame with Ethernet Type = 0x8892. Thus, for the cyclic IO-Data, the IP address is not used anymore. Only MAC addresses are used.

## 3. Possibilities of Errors

With such a setup sequence, different assignments can fail and result in errors. We will investigate the two most interesting cases for our study, which are when there are duplicate assignments of Profinet names or IP addresses amongst the IO-Devices.

### 3.1. Duplicate assignment of Profinet names

During the planning phase of an installation we are free to assign the names of the devices as we like. The engineering tool typically suggests default names. During the engineering phase these names need to be assigned to the real IO-Devices over the network. This is done by the DCP Protocol. In the initial multicast all Profinet devices are ask to specify the status of their names. This list is presented to the user, who selects, based on the MAC address as unique identifier, the IO-Device and gives the command to assign the name to the IO-Device. This assignment is done by a DCP-Set frame. It is possible to assign the same name to different devices. During the setup of the AR, the IO-Controller tries to identify the IO-Device with the specified name with the DCP-Identify frame. If there are multiple replies, the protocol specification halts further processing and the AR setup fails with the user receiving an error message.

### 3.2. Duplicate assignment of IP addresses

If the assigned Profinet name is unique, the IO-Controller verifies the IP address to be unique with an ARP request. The standard ARP communication stack does not handle double assigned IP addresses correctly. If there is more than one reply to an ARP request, the ARP table is updated with the last one received. Thus, if there are multiple devices with duplicate IP addresses, only the last reply will be used in the ARP table. The elapsed time for the IO-Device's reply depends on the topology and delays in your network. Typically, there is no signaling of the ARP software to the upper layers that there are multiple devices signaling the same IP address.

It was possible to reproduce this error in our lab with commercial Profinet IO products with varying results. The setup consisted of a PC based IO-Controller from Siemens running on a normal PC and two simple IO-Devices with a built-in switch from Phoenix Contact. Independent of the Profinet name, it was possible that either of the two IO-Devices would have an AR established with the IO-Controller, depending on which device replied last. However, if the connection setup begins between the replies of the two IO-Devices, the setup will be unexpectedly interrupted by the second IO-Device's reply. This will result in a protocol error.

This incorrect setup of the AR is only possible if the two IO-Devices are of the same type. On the DCE RPC setup there is a type-code included, to verify the correct type of the device.

An attempt was made to reproduce this error with an IO-Controller implemented in a Programmable Logic Controller (PLC), a Siemens S7-315-2 DP/PN. The IO-Controller detects the duplicate IP address assignments and refuses to initiate the AR with the IO-Device. This is done according to [3] by an ARP_Timer after the transmission of the ARP request. If an ARP reply is sent this should be signaled by the ARP stack to the PROFINET Name Resolution Protocol Machine (NRPM) and the error is detected. If an IO-Controller is hosted on a commercial Operating system like Windows it may not be possible to force the ARP stack to signal this occurrence of duplicate IP address replies. Thus, the duplicate assignment of IP addresses remains undetected.

### 3.3. Mix-up the MAC addresses

Typically the MAC address is unique for one device in the world. But today, there are different sniffer and hacking tools, which allow replacing and modifying MAC address of the device with ease. It is interesting to reflect on the possibilities of these methods.

The MAC address is used by an Ethernet switch device to determine which port to forward an Ethernet frame to. For this, the switch learns the sender's MAC address from the incoming messages. A device in an IT network may move around and be connected to different ports on a switch. Therefore the switch is required to continuously learn the new MAC address assignments to the ports.

The principle operation of a man-in-the-middle (MITM) attack tool is described as follows. The MITM sends an Ethernet frame to the switch with a foreign MAC address and the switch forwards all frames for this device to the MITM instead of sending the frames to the dedicated device. In order that the faked device is not timed out, the MITM sends the monitored frame forward again to the original destination.

There is no reason, why this MITM attack would not work also for a Profinet IO system. The main condition is that standard commercially available Ethernet switches are used (refer to Figure 1).

There are some timing constraints to be followed by such an MITM attack. For every frame sent by the IO-Device to the switch, the switch updates its MAC table. Thus, the MITM must immediately send another fake frame to change the entry in the MAC table of the switch. If the cyclic IO-Data of the IO-Device is sent every 1 ms to the IO-Controller, the MITM must also send a fake message every 1 ms. It the MITM forwards the IO-Data frame to the IO-Controller inside the cycle time this condition is fulfilled. The same applies to the address of the IO-Controller. This generates a double load on the MITM port compared to the load on the other ports of the switch.

## 4. Possibility of a device emulator

Is it possible to use an open source MITM attack tool or to build a dedicated system.

### 4.1. Using ettercap

We attempted to use an open-source MITM attack tool [4] to get access to a Profinet IO-System. But there was no success due to the following reasons:

The function "ARP poison" sends cyclic ARP frames to the target system to confuse the ARP table in the target stations. For devices on a Profinet IO system this is only of importance for the AR setup. The cyclic data always resets the MAC addresses in the switch. We reduced the cycle time of the IO-Data to 64 ms, but the ettercap tool has its smallest cycle with 1 second.

If we attempt the "Port stealing" attack, the ettercap program sends a gratuitous ARP message to all target devices in the network every second. But the cyclic data is faster and the switch always regains the correct MAC address. The standard ettercap tool supports TCP port stealing, and is not prepared for UDP transmissions.

This attack is more successful in the "bridged" mode instead of the "unified" mode (refer to Figure 4). In the bridged mode a PC with two Ethernet ports is used and connected inside the Ethernet connection to be sniffed.

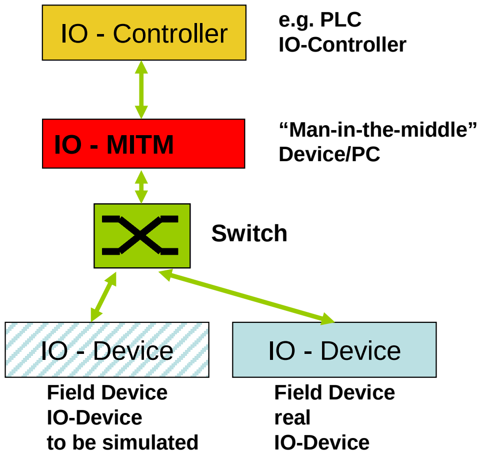

Figure 4: MITM "bridged" mode

### 4.2. Dedicated tool

Besides the possibility to use a sniffer in a switched Profinet IO network there are other possibilities to use the following properties of an Ethernet network.

In a Profinet IO installation there is sometimes the need to test and control a program in an IO-Controller with wrong input data. We should now be able to construct an IO-Device emulator describes as follows.

The IO-Controller is faked with the IP address or the connected switch is faked with the MAC address of the IO-Device to be emulated. The IO-Controller still has the impression that it is connected to the correct device, but in fact the frames are sent to the IO-Device emulator. There is even no need that the emulator sends the data to the disconnected IO-device. With this method it is possible to emulate input values without even disconnecting the real device from the network.

If the device emulator is built as a bridged device, it passes all frames through to other devices but replies to the frames addressed to the IO-Device to be emulated.

## 5. Outlook

It will be interesting to see the possibilities to fake other Ethernet based industrial networks. It is assumed, that there are the same possibilities as long as the standard functionality of a switched Ethernet is not modified.

### References

[1] J.Feld: PROFINET – Scalable Factory Communication for all Applications, 2004 IEEE International Workshop on Factory Communication Systems, September 22 – 24, 2004, Vienna, Austria, page 33 – 38

[2] IEC/PAS 62411 Ed.1: Real-Time Ethernet PROFINET IO, Publicly Available Specification for Real-Time Ethernet, Proposal available as 65C/359/NP, Date of circulation: 2004-12-03

[3] PROFIBUS International: Protocol and Services - Input for IEC 61158 Ed. 4, Document number 7.032, Version 0.9, November 2005, available at www.profinet.com

[4] http://ettercap.sourgeforce.net

[5] Profibus International: PROFINET: Technology and Application, System Description, Document number: 4.132, Issue April 2006, available at www.profibus.com

[6] Felser, M.: Real-Time Ethernet - Industry Prospective, PROCEEDINGS OF THE IEEE, VOL 93. NO.6, JUNE 2005, Pages 1118 ff