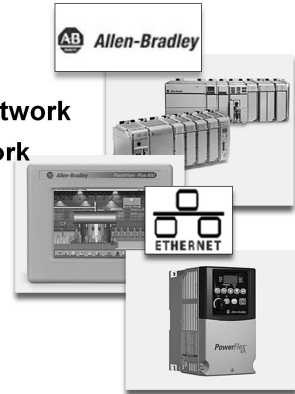




CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory:
Description of laboratory and basic scenario
- 6 Software tools



Co-operative Network Training



CoNet Mobile Lab: EtherNet/IP on Allen Bradley platform

Introduction

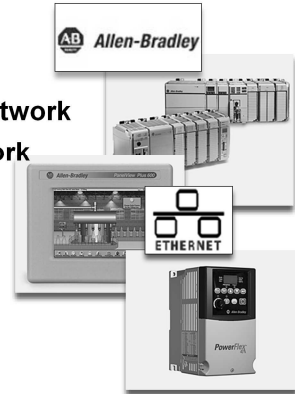
- 1 Distributed Control Architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory: Description of laboratory and basic scenario
- 6 Software tools

© 2011 Wojciech Grega, Department of Automatics, AGH University of Science and Technology



CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture**
- 2 Real-time control system and real-time network**
- 3 Monitoring and testing the Ethernet network**
- 4 Introduction to EtherNet/IP technology**
- 5 Introduction to laboratory:
Description of laboratory and basic scenario**
- 6 Software tools**



CoNet Mobile Lab: EtherNet/IP on Allen Bradley platform

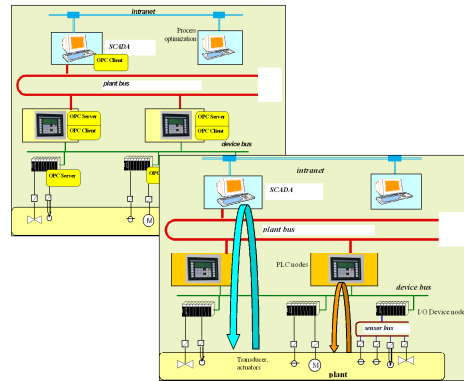
Introduction

- 1 Distributed Control Architecture**
- 2 Real-time control system and real-time network**
- 3 Monitoring and testing the Ethernet network**
- 4 Introduction to EtherNet/IP technology**
- 5 Introduction to laboratory: Description of laboratory and basic scenario**
- 6 Software tools**

<1 Distributed control architecture>

1.1 Structure of the industrial control system

1.2 Integration problem



© 2011 Wojciech Grega, Department of Automatics, AGH-UST



Content of the lesson „EtherNet/IP on Allen-Bradley platform”

1.1 Structure of the industrial control system

1.2 Integration problem



Structure of Industrial Control Systems

There are three major trends in contemporary industrial systems:

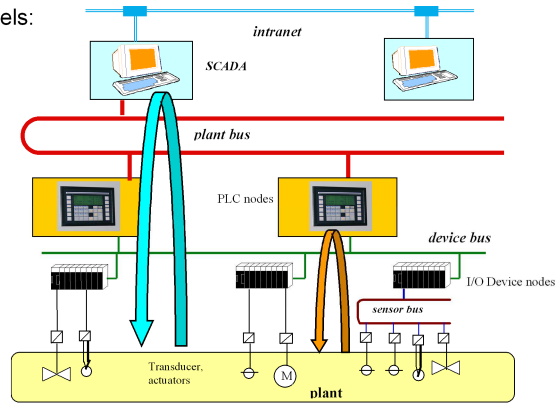
- Distributed and decentralized structures of automation,
- Increased integration of communication through all levels of the control systems supported by the application of wired and wireless networks,
- Growing demand for the application of IT standards.

Most industrial control systems adopt a **multilevel, vertical control hierarchy**.

Such a system is structured in three levels:

- the direct (device) control level,
- the supervisory level,
- the management level.

Communication networks at each level ensure data transmission and cooperation between distributed control nodes.



© 2011 Wojciech Grega, Department of Automatics, AGH-UST



<1 Distributed Control Architecture>
<1.1 Structure of the industrial control system>

Currently, there are three major trends in contemporary industrial control systems:

- Distributed and decentralized structures of automation,
- Increased integration of communication through all levels of the control systems supported by the application of wired and wireless networks,
- Growing demand for the application of IT standards.

Fig.1.1 multilevel structure of an industrial control system

Most industrial control systems adopt a multilevel, vertical control hierarchy. Logically, such a system (see fig. 1.1) is structured in three levels: the direct (device) control level, the supervisory level and the management level.



Flow of Information Supported by Network

- Control (synchronous)
 - **Implicit** messaging (I/O messages that typically contain time-critical control data):
 - o acquisition of process input data (synchronous)
 - o actuation of process outputs
- Collect (asynchronous)
 - **Explicit** Messaging:
 - o transmission of data between controllers
 - o setting of parameters by supervisory systems
- Configure
 - Initial Commissioning
 - Program Upload/Download
 - On-line enhancements and modification

Before we look at how ODVA & CI have built EtherNet/IP, we must first look at the three principal types of functionality offered to the user.

Control services provide for the traditional I/O of a PLC based system. The I/O of the system is predictable, in so far as the presentation of data will follow a preset, pre-configured pattern as far as is practical with no user input. Hence implicit; by specifying the presence of I/O devices, the data exchange messaging to them is implicitly specified also with no further need for the user to interact with the application layer

Collection services provide for the traditional monitoring of a control system parameters and databases and the non-time critical interfacing between them. Messaging is asynchronous in that when a message is transmitted or received there is no guarantee about when or even if the same message will be transmitted or received again. Each message must be explicitly triggered by a call into the application layer of the protocol

Finally, configuration services provide for the system being set up initially, or modified on-line. They allow the application itself to be modified, rather than the application data which is all that is made available to the Collection services

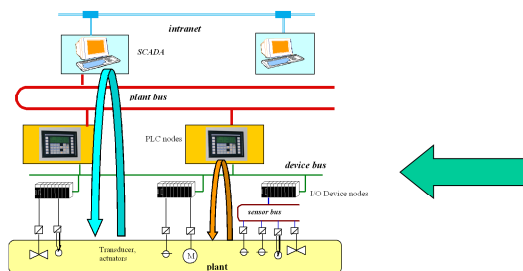
Direct (device) Control

MAIN TASK - maintain the process states at the prescribed set values

Level components and its functionality

- Interface to the hardware
- Control algorithms (mainly PID)
- PLC and embedded control nodes as the front-ends of the system
- High-quality networks (speed, punctuality)

Hard real-time activities are located at this level



© 2011 Wojciech Grega, Department of Automatics, AGH-UST

<1 Distributed Control Architecture>
<1.1 Structure of the industrial control system>



The basic task of the direct (device) control level is to maintain the process states at the prescribed set values. The device controller level provides an interface to the hardware, either separate modules or microprocessors incorporated in the equipment to be controlled. Here, mainly PID digital control algorithms are implemented – in some cases these are more advanced control methods such as multivariable control or adaptive functions. A number of embedded control nodes and Programmable Logical Controllers (PLC) are used as the front-ends to take the control tasks. High speed networks and fieldbuses are implemented at the direct control level to exchange in real time the information between front-ends and the device controllers and, vertically, with the supervisory control level. This architecture has the advantage of locating the hard real-time activities as near as possible to the equipment.



Supervisory Level

The supervisory level comprises workstations and industrial PCs providing high-level control support, database support, graphic man-machine interface, network management and general computing resources.

Main Tasks:

- Calculates set points for controllers according to the defined criteria,
- Solves optimization and identification tasks,
- Uses more advanced closed-loop control strategies (predictive control, repetitive control),
- Offers shorter computational time due to the higher efficiency of the workstations,
- Provides redundant control system if necessary (control loops and algorithms)



The supervisory level comprises workstations and industrial PCs providing high-level control support, database support, graphic man-machine interface, network management and general computing resources.

Classically, the supervisory level calculates set points for controllers according to the defined criteria. For this purpose more complex mathematical models of the process are employed at this level to find the optimal steady-state, by solving optimisation and identification tasks. Due to the rapid development of computer technology, there is growing scope for more advanced close-loop algorithms (predictive control, repetitive control) located at this level. However, increasing computational efficiency of PLCs at the device level supported by high performance networks transferring data and control signals vertically gives more flexibility to the designer. The control loops can be handled by local, device-level controllers, and also by the supervisory controllers (Fig.1.1). For example, a predictive control algorithm can be handled by a supervisory workstation as well as by a local PLC. In some cases similar control algorithms must be located in both levels if redundancy of the control system is required. It should be noted that upper level loops usually offer shorter computational time due to the higher efficiency of the workstations.

The role of the communication networks at each level is to ensure data transmission and coordinate manipulation among spatially distributed control nodes. The evolution of industrial communication has moved to Industrial Ethernet networks [1]. Since Ethernet is a shared network, the packets containing the digitized measurements need to share the network bandwidth with external traffic; thus, the available channel capacity is limited and dynamically changing. Therefore, the control over the upper level loop usually offers longer data transfer time due to high vertical network traffic and longer delays.

Integration

INFORMATION INTEGRATION IS A CRITICAL ISSUE

Common industrial communication protocol and integration standard to facilitate interoperability needed!

Open and effective communication and integration architecture focused on data access.

Ethernet protocol – **data transmission standard**

OPC (OLE for Process Control) – **data exchange standard** providing open data integration architecture. Between nodes and between software modules



The ability to easily integrate information from control systems and „plant floor” measurements with supervisory and optimisation levels is a critical issue. It is very difficult to share data between various devices and software manufactured by different vendors without a common industrial communication protocol and integration standard to facilitate interoperability.

The key is an open and effective communication and integration architecture concentrating on data access, not on the type of data. Ethernet protocol comes as a solution to the data transmission problems mentioned above, while the OPC (OLE for Process Control) data exchange standard was developed as a solution which fulfills the requirements of open data integration architecture.



OPC



OPC:

- Based on the Microsoft DCOM specification
- Most commonly used form is DATA Access (OPC DA) – deals only with current process data
- OPC client can be connected to several OPC servers provided by different vendors
- OPC client can connect to the OPC server which is located in another computer in a network (DCOM)

OPC – implements client-server communication

- ‘cache’ or ‘device’ operations
- synchronous or asynchronous reading and writing

The OPC standard was not primarily intended for feedback control – one of the most important issues for communication between devices is proper configuration and synchronization.



OPC is based on the Microsoft DCOM specification. Although OPC actually consists of many different data exchange specifications, its most commonly used form is Data Access (OPC DA), which supports both client-server and publisher-subscriber data exchange models. OPC DA deals only with current process data - not historical data or alarms.

An OPC client can be connected to several OPC servers provided by different vendors. An important feature of OPC is that the client is able to connect to the OPC server which is located in another computer in the network. This is possible because of DCOM (Distributed COM) the COM standard extension which enables a client running on one computer to create instances and invoke methods of servers on another computer within the network.

The OPC standard defines numerous ways to communicate between the server and its clients to satisfy different OPC applications. The client can specify that some operations should be performed on “cache” or “device”. If “device” is chosen, operation directly on the physical device will be requested. If “cache” is selected data will be read from the server’s internal memory where the server keeps a copy of the device data received during the last OPC refresh cycle.

Both reading and writing can be done synchronously or asynchronously:

- OPC synchronous functions run to completion before returning. This means that the OPC client program execution is stopped till operation is completed.
- OPC asynchronous functions use a callback mechanism to inform the requested OPC operation is over. The client program execution is not stopped while waiting for OPC operation to be completed.



OPC



© CoNeT - Co-operative Network Training

OPC and the control architecture

- Enables exchange data between
 - Multi-vendors devices and control software (horizontal integration)
 - Different software applications (vertical integration) see Fig. 1.2

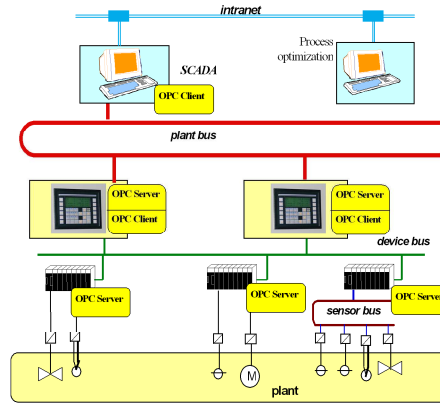


Fig. 1.2 Vertical integration implementing the OPC data exchange model

© 2011 Wojciech Grega, Department of Automatics, AGH-UST

<1 Distributed Control Architecture>
<1.2 Integration problem>

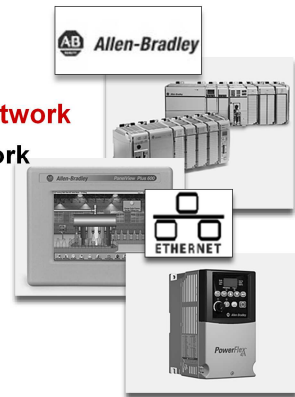


OPC is a widely accepted open industrial standard that enables the exchange of data without any proprietary restrictions between multi-vendor devices and control software (horizontal integration) as well as between different software applications (vertical integration, Fig.1.2). Currently, most plant-level control systems can be configured to be the OPC servers for supervisory control levels of the factory.



CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory:
Description of laboratory and basic scenario
- 6 Software tools



Co-operative Network Training



CoNet Mobile Lab: EtherNet/IP on Allen Bradley platform

Introduction

- 1 Distributed Control Architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory: Description of laboratory and basic scenario
- 6 Software tools

© 2011 Wojciech Grega, Department of Automatics, AGH University of Science and Technology

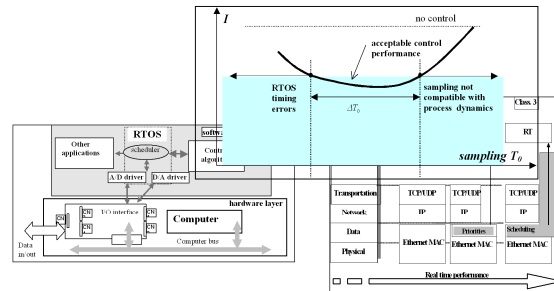
<2 Real-time Control System and Real-time Network>

2.1 Real-Time and Control

2.2 Computer Implementations of Real-Time Control Systems

2.3 Real-Time Distributed Control Systems

2.4 RTOS and Control System Performance



© 2011 Wojciech Grega, Department of Automatics, AGH-UST



Content of the lesson „EtherNet/IP on Allen-Bradley platform”

2.1 Real-Time and Control

2.2 Real-Time Distributed Control Systems

2.3 Computer Implementations of Real-Time Control Systems

2.4 RTOS and Control System Performance



Computer-based control systems

Requirements on the hardware and software:

- Sufficient processing power,
- Sufficient high-speed input/output interfaces (peripheral hardware),
- Hard real-time requirements (met more or less),
- Handling error conditions in a predefined way,
- Control system reliability (hardware durability and operating system stability) in safety-critical applications

Real-time: The operating mode of a computer system in which the programs for the processing of data arriving from the outside are permanently ready, so that their results will be available within predetermined periods of time; the arrival times of the data can be randomly distributed or be already determined depending on the different applications.

Computer based digital controllers typically have the ability to monitor a number of discrete and analog inputs, perform complex control algorithms, and drive several outputs, all at defined speeds, often very high. It is necessary that all the above operations and calculations take place at the correct moment in time. This imposes the following requirements on the hardware and software of computer-based control systems:

- sufficient processing power,
- sufficient high-speed input/output interfaces (peripheral hardware),
- operating systems fulfilling more or less hard real time requirements and handling error conditions in a predefined way.

In many cases the processing power of digital controllers is not the only priority. Some applications are classified **as safety-critical**, in the sense that computer failure may result in incorrect or disastrous behaviour of the process (for instance a threat to life, or to the Earth's environment). In this case, the list of the requirements must be extended to include control system reliability (hardware durability and operating system stability).

Real-time control systems

Only a proper distribution of computer resources together with sufficient computational power allows one to build a predictable Real Time Control System (RTCS).

Real-time systems classification:

- Hard real-time systems,
- Soft real-time systems,
- Firm real-time systems.

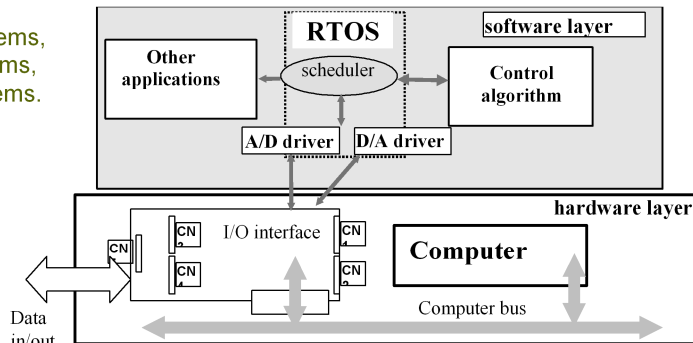


Fig.2.1. Hardware and software layers of the computer controlled system

© 2011 Wojciech Grega, Department of Automatics, AGH-UST

<2 Real-time control system and real-time network>
<2.1 Real-Time and Control>



In general, a computer-based digital controller must detect real-world events and respond to them by taking appropriate actions. This feature of the controller is referred to as **real-time operation: the operating mode of a computer system processing asynchronous inputs and producing outputs in a deterministic and bounded amount of time**. The arrival times of the data can be randomly distributed or can be determined depending on the different applications. A real-time system is one in which the correctness of the system operation depends not only on the logical results of computation, but also on the time at which the results are generated. Hardware over-capacity may satisfy the execution duration of all the applications but may not satisfy the predictability. Only a proper distribution of computer resources together with sufficient computational power allows one to build a predictable Real Time Control System (RTCS). Real-time systems are usually classified into hard real-time systems, soft real-time systems and firm real-time systems.

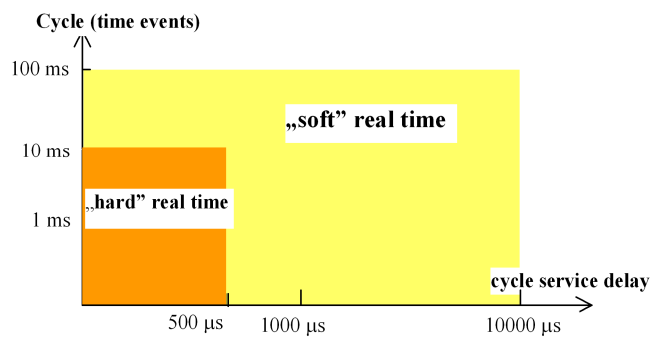
- A **hard real-time system** is a system, where missing the response deadline can be catastrophic.
- A **soft real-time system** is a system where deadlines are important but where the system operates properly if the deadlines are occasionally missed (system performance is degraded, but not destroyed).
- The term **firm** relates to a real-time system where late data processing results are worthless and some probability of violating a response time is tolerable.

A typical control system consists of: a controlled plant, sensors, actuators, input-output interfaces and a controller including a Real-Time Operating System (RTOS). There are three basic concurrent processes of RTOS: performing measurements, data processing and running control algorithms.

Fig.2.1 illustrates the principle of co-operation between hardware and software layers. The RTOS operates on device drivers. A **driver** is a software "image" of the I/O hardware. Access to the I/O devices is possible by device drivers. The RTOS synchronises the information flow between the hardware level and control applications.

Real-time control systems

- A **hard real-time system** is a system, where miss of response deadline can be catastrophic.
- A **soft real-time system** is the system where deadlines are important but where the system operates properly if the deadlines are occasionally missed (system performance is degraded, but not destroyed).
- The term **firm** relates to the real-time system, where late data processing results are worthless and some probability of violating a response time is tolerable.



© 2011 Wojciech Grega, Department of Automatics, AGH-UST



<2 Real-time control system and real-time network>
<2.2 Computer Implementations of Real-Time Control Systems>



Real-time Control Computers with RTOS

Real-time control computers with RTOSs are applied as:

- Embedded Systems, when the computer becomes a component of a larger system,
- Industrial Control Systems, when the computer creates a self-contained control configuration.

Table 2.1. Industrial RT solutions

Hardware platform	Minimal cycle ^{*)}	Examples of RTOS
Microcontrollers	10- 50 μ s	-
DSP controllers	1 μ s	dSPACE
PLC	1-20 ms	dedicated
FPGA controllers	10 ns	-
FPGA controllers	10 μ s	EDA
VME industrial controllers	100 μ s	QNX, VxWorks
IPC ^{*)}	100 μ s	Extended Windows, RTLinux

^{*)} industrial PC, ^{**) time interval between input reading and output signal generation}

© 2011 Wojciech Grega, Department of Automatics, AGH-UST



<2 Real-time control system and real-time network>
 <2.2 Computer Implementations of Real-Time Control Systems>

Real-time control computers equipped with RTOSs are applied as:

Embedded Systems, when the computer becomes a component of a larger system:

- microcontrollers based on dedicated processors,
- miniature WEB servers hosting I/O circuits,
- controllers based on software-configurable FPGA technology, e.g. XILINX chips.

Industrial Control Systems, when the computer creates a self-contained control configuration:

- Programmable Logic Controllers (PLCs), soft-PLCs,
- open-standard industrial computers (e.g. VME or PC104 bus industrial computers holding appropriate I/O interfaces),
- distributed control systems, organized according to some hierarchy, industrial computers (IPC) supervising a more or less intelligent input/output boards.

Most embedded systems are dedicated. This means that their functionality is tied to hardware and software for ever. They are widely used in everyday electronic equipment, e.g. transaction systems or multimedia equipment. They fulfil the essential requirement of a real-time application such as a deterministic response time and operating system stability. However these solutions are not flexible.

In industry, PLCs are still a standard control solution for both continuous and sequential processes. The VME industrial computers offer hardware and software applications for harsh conditions. They are flexible i.e. the computer is used as a development and implementation platform. Their main drawback is the price. The guaranteed cycle times of RTOS applied to different hardware platforms is shown in Table 2.1.



Real-time network

The operating mode of a data transmission network in which the data arriving from the outside nodes are permanently ready, so they are available within predetermined periods of time.

Communication Networks:

- Fieldbuses,
- General Purpose Networks.

Advantages:

- Low installation cost,
- Ease of maintenance,
- Flexibility

Time delays (how to deal with them?):

- Communication protocols minimising their impact,
- Consideration of the distributed control system as a real-time system,
- Control algorithm dedicated to systems with time delays

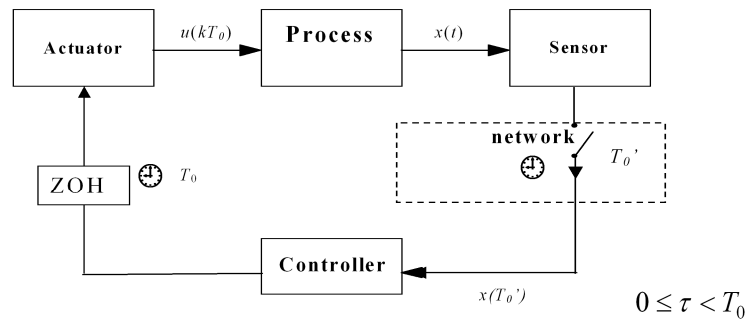
Feedback control systems wherein the control loops are closed through a communication network are referred to as distributed control systems. They are distributed in the sense that their sensors, actuators and controllers communicate via a shared data transmission network. Communication networks were introduced in control in the 1970s. They can be grouped into fieldbuses (e.g. CAN, Profibus, Modbus) and general purpose networks (e.g. IEEE standard LANs). Each type of network has its own protocol that is designed for a specific range of applications. Fieldbuses are intended for real-time applications, but in some cases general-purpose networks may have been used for control [3]. The behaviour of a networked control system depends on the performance parameters of the underlying network, which include transmission rate and access method to the network transmission medium.

Using a distributed control architecture has many advantages over a point-to-point design including low cost of installation, ease of maintenance and flexibility [4]. The introduction of distributed architecture can improve the reliability and efficiency of the control application. For these reasons distributed control architecture is widely used in industrial applications.

The introduction of a network into a feedback loop in some cases violates conventional control theory assumptions such as non-delayed sensing and actuation. These time delays mainly come from the time sharing of the communication network. **Lack of access to the communication network is an important constraint compared to lack of computer power or time errors of the RTOS.** Time delays can degrade the performance of the control system designed without considering network delays and can even destabilise the system.

Communication protocols that minimise data delay and make the system time invariant are widely introduced. There are other approaches for accommodating all network effects in control system design. One way is to treat a distributed control system as a real-time system, where data transfer from/to the plant and control of communication lines are real-time tasks. For such a task one can use all the standard methods for real-time system design: define priorities for task scheduling methods and algorithms and time constraints etc. Finally, a distributed RTOS system is a design which enables the flow of information in a limited amount of time.

Distributed control systems



Introduction of an „office” Ethernet network into control loop can degrade the performance of the control system. The main complication of this control architecture is the presence of variable *time delays*. Other complicated situation occur if *samples are lost* during transmission.

© 2011 Wojciech Grega, Department of Automatics, AGH-UST

<2 Real-time control system and real-time network>
<2.3 Real-Time Distributed Control Systems>





Ethernet: about determinism

Popular solution in the area of industrial communications.

In industrial environment its **deterministic operation** can be assured by:

- Keeping network load low
- Segmenting the network using switches (IEEE 802.1D) – Avoids collisions
- Embedding a fieldbus or application protocol on TCP/IP
- Using a special Data Link layer for real-time devices
- Using application protocol on TCP/IP, direct MAC addressing with prioritization for real-time, and hardware switching for fast real-time,
- Maintaining real-time on TCP/IP is achieved by prioritized messaging and time synchronization.

Current communication systems for automation implement different protocols. This is a substantial disadvantage, leading to the need to use vendor-specific hardware and software components, which increase installation and maintenance costs. Moreover, presently used fieldbus technologies make vertical communication across all levels of the automation systems difficult. Gateways need to be used to establish connections between different kinds of fieldbus systems used in the lower levels, and Ethernet used in the upper levels.

Ethernet provides unified data formats and reduces the complexity of installation and maintenance, which, together with the substantial increase in transmission rates and communication reliability over the last few years, results in its popularity in the area of industrial communications.

Ethernet, as defined in IEEE 802.3, is non-deterministic and, thus, is unsuitable for hard real-time applications. The media access control protocol, CSMA/CD with its backoff algorithm, prevents the network from supporting hard real-time communication due to its random delays and potential transmission failures. In real-time systems, delays and irregularities in data transmission can very severely affect the system operation. Therefore, various techniques and communication protocol modifications are employed in order to eliminate or minimise these unwanted effects.

To employ Ethernet in an industrial environment, its deterministic operation must first be assured. This can be accomplished in several ways. Coexistence of real-time and non-real time traffic on the same network infrastructure remains the main problem. This conflict can be resolved in several ways by:

- embedding a fieldbus or application protocol on TCP/IP – the fieldbus protocol is tunneled over Ethernet, and full openness for “office” traffic is maintained,
- using a special Data Link layer for real-time devices – special protocol is used on the second OSI Layer, implemented in every device. The real-time cycle is divided into slots, one of which is opened for regular TCP/IP traffic, but the bandwidth available is heavily limited (i.e., minimized),
- using application protocol on TCP/IP, direct MAC addressing with prioritization for



Ethernet

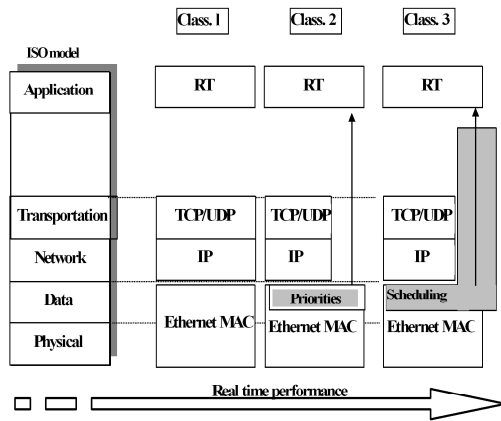
- **Ethernet-based control** implies temporal non-determinism:
 - network communication
 - real-time scheduling
- Degraded control performance due to:
 - sampling interval jitter
 - non-negligible input-output latencies with jitter
 - lost samples
- However:
 - Most control loops are fairly robust towards temporal non-determinism



Ethernet



Real-time Ethernet - PROFINET, EtherCAT, Ethernet/IP and many more..



Generally conformance with the Ethernet standard decreases when ones increase the Class number, while the achievable realtime performance increases.

Fig.2.2 Classification of industrial Ethernet (IEC 61 784-2)

© 2011 Wojciech Grega, Department of Automatics, AGH-UST

© CoNeT - Co-operative Network Training

<2 Real-time control system and real-time network>
<2.3 Real-Time Distributed Control Systems>



The desire to incorporate a real-time element into this popular single-network solution has led to the development of different real-time Industrial Ethernet solutions, called Real-time Ethernet, such as PROFINET, EtherCAT, Ethernet/IP and many more. The conditions for the industrial use of Ethernet are described by international standard IEC 61 784-2 *Real Time Ethernet* (See Fig.2.2). IEC stands for *International Electrotechnical Commission*.

Class 1 describes the use of standard Ethernet TCP/IP as it is. In this case the different real time protocols and the best-effort protocols, like HTTP, SNMP, FTP etc., uses the services of the TCP/IP protocol suite. This includes examples such as CIP Sync (Ethernet/IP, ModBus/TCP). The class 1 has the largest conformity to the Ethernet TCP/IP standard and can thereby use standard hardware and software components.

Class 2 introduces optimizations, whereby the realtime data bypasses the TCP/IP stack and thus considerably reduces the dwell time in the node and increases the achievable packet rate. The dwell time of the node is one of the substantial influence factors for the realtime performance and has for embedded devices typical values of 1ms. In Classes 1 and 2, the priority support described in IEEE 802.1Q can also be used depending on the approach. In Class 3 the scheduling on the MAC level is again modified through the introduction of a TDMA method. Class 3 can be used in applications that require maximum latency in the range 1ms and a maximum jitter of < 1micros. In this class there are strong restrictions for the use of standard components or the necessity for special components, like switches. Generally conformance with the Ethernet standard decreases when ones increase the Class number, while the achievable realtime performance increases.

Overview of Technologies

© CoNet - Co-operative Network Training

- CC Link IE
- Drive CliQ
- EPA
- EtherCAT
- EtherNet/IP + CIP Sync
- Ethernet Powerlink
- IEEE 1588 / PTP
- JetSync
- Modbus RTPS
- Mechatrolink III

- PowerDNA
- Profinet
- RAPIEnet
- RTEX
- SafetyNET p
- SERCOS III
- SynqNet
- TCnet
- IEC61850
- Vnet/IP

<2 Real-time control system and real-time network>
<2.3 Real-Time Distributed Control Systems>

Source: <http://www.real-time-ethernet.de>

No. 22 

© CoNet - Co-operative Network Training


Overview of Technologies

- EPA
- EtherCAT
- EtherNet/IP + CIP Sync
- Ethernet Powerlink
- IEEE 1588 / PTP
- Modbus RTPS
- Profinet
- RAPIEnet
- SERCOS III
- Tcnet
- IEC61850
- Vnet/IP

International Standard

<2 Real-time control system and real-time network>
<2.3 Real-Time Distributed Control Systems>

Source: <http://www.real-time-ethernet.de>

No. 23 



© CoNet - Co-operative Network Training


Overview of Technologies

- EtherCAT
- EtherNet/IP + CIP Sync
- Ethernet Powerlink
- Modbus RTPS
- Profinet
- SERCOS III
- Tcnet
- IEC61850
- Vnet/IP

User Organization

<2 Real-time control system and real-time network>
<2.3 Real-Time Distributed Control Systems>

Source: <http://www.real-time-ethernet.de>



© CoNeT - Co-operative Network Training

Overview of Technologies

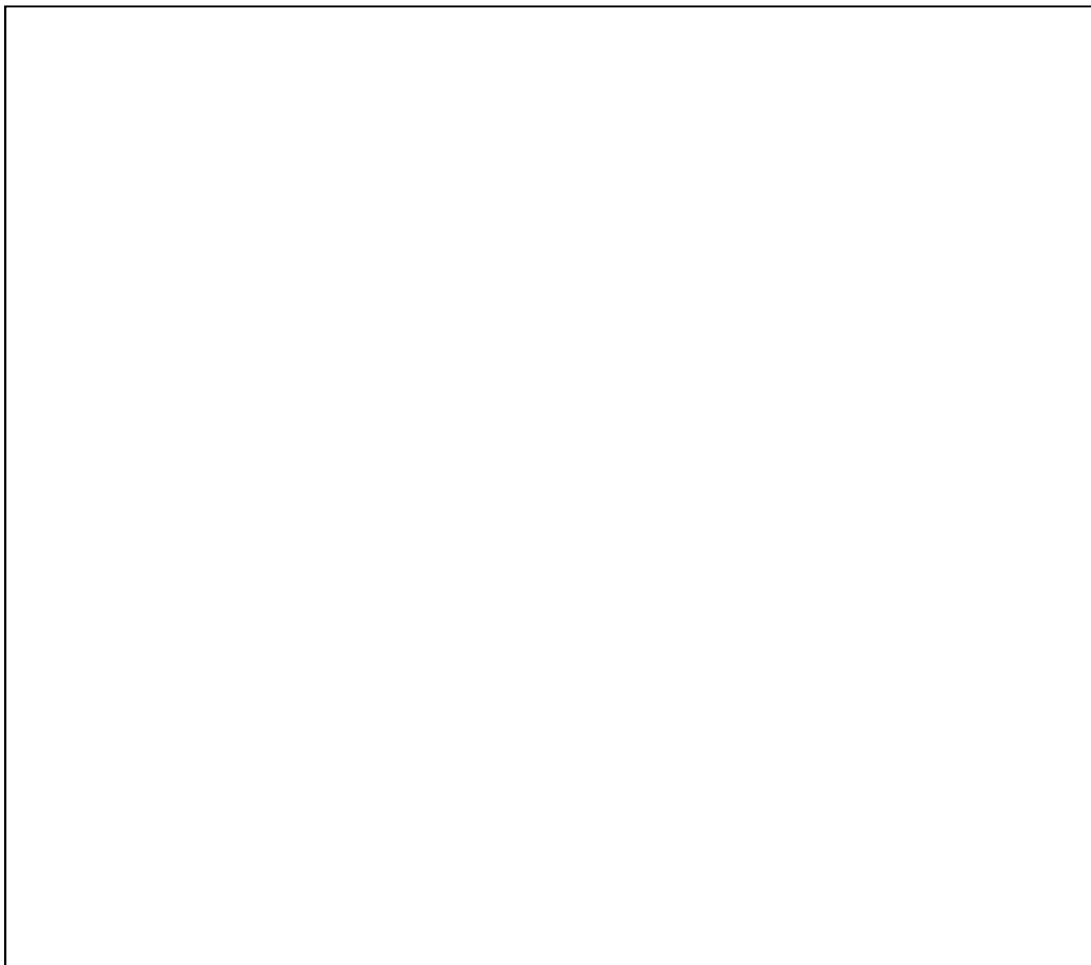

- EtherCAT
- EtherNet/IP + CIP Sync
- Ethernet Powerlink

- Profinet
- SERCOS III
- IEC61850

Performance

<2 Real-time control system and real-time network>
<2.3 Real-Time Distributed Control Systems>

Source: <http://www.real-time-ethernet.de>





Ethernet Performance

Ethernet performance metrics:

- Latency (delay)
- Jitter
- Loss rate
- Throughput
- Error rate
- Bit error rate

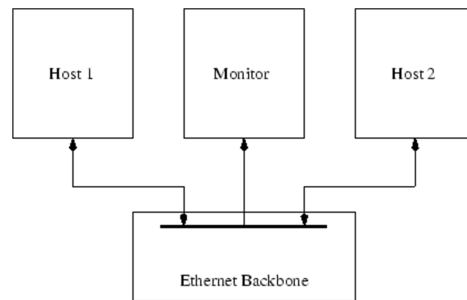


Fig 3.1. Basic Ethernet measurement setup

Software and hardware applications to measure the Ethernet performance.

The following parameters are covered by the Ethernet performance metrics:

- **Latency** (delay) – the amount of time required for a frame to travel from source to destination.
- **Jitter** – a measure of the deviation of the latency from its average value.
- **Loss rate** – the probability that an individual packet is lost (dropped) during the transmission.
- **Throughput** – the amount of digital data transferred per time unit.
- **Error rate** – the ratio of the number of erroneous units of data to the total number of units of data transmitted.
- **Bit error rate** – the ratio of the number of incorrectly received bits to the total number of transmitted bits.

There are many free and commercial applications available for common Operating Systems such as MS Windows, Linux and OS X that can provide varying LAN throughput performance measurement. There are also self contained hardware based solutions that provide the most accurate LAN performance measurement / stress testing.

Software applications can cost up to several hundred euros. Hardware based solutions range from a few hundred to tens of thousands of euros and are typically used by manufacturers, carriers and professional IT consulting organizations.

In its most basic configuration Ethernet is relatively simple to measure. The base configuration is shown in Figure 3.1.



Definitions

Jitter is an unwanted variation of one or more characteristics of a periodic signal. Jitter may be seen in characteristics such as the interval between successive pulses, packets or tasks. Jitter is a significant factor in the design of almost all communications links or RT systems.

Latency is a time delay between the moment something is initiated, and the moment one of its effects begins or becomes detectable. The word derives from the fact that during the period of latency the effects of an action are latent, meaning "potential" or "not yet observed". Even within an engineering context, latency has several meanings depending on the engineering area concerned (i.e. communication, operational, simulation, mechanical, or biomedical fiber stimulation latencies).

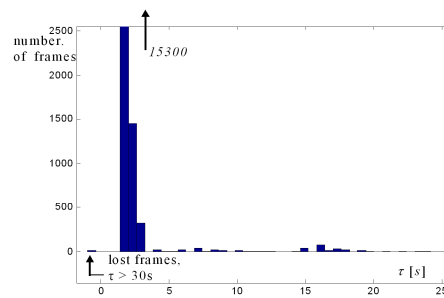
© 2011 Wojciech Grega, Department of Automatics, AGH-UST



<2 Real-time control system and real-time network>
<2. 4 RTOS and Control System Performance>

Why is Jitter Bad?

- The control algorithms are designed assuming a constant T_0
- The jitter can be interpreted as a process disturbance
- Very difficult to analyze in the general case
- The Jitterbug Toolbox or True Time Toolbox can be used to evaluate the effect of jitter for a given case
- Many jitter compensation schemes have been developed



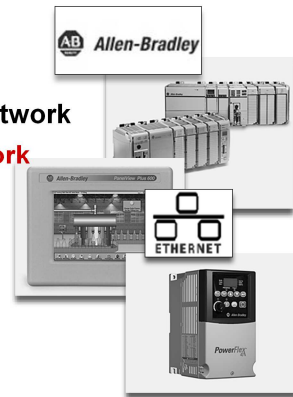
© 2011 Wojciech Grega, Department of Automatics, AGH-UST



<2 Real-time control system and real-time network>
<2. 4 RTOS and Control System Performance>

CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture
- 2 Real-time control system and real-time network
- 3 **Monitoring and testing the Ethernet network**
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory:
Description of laboratory and basic scenario
- 6 Software tools



Co-operative Network Training



CoNet Mobile Lab: EtherNet/IP on Allen Bradley platform

Introduction

- 1 Distributed Control Architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory: Description of laboratory and basic scenario
- 6 Software tools

© 2011 Wojciech Grega, Department of Automatics, AGH University of Science and Technology



<3 Monitoring and testing the Ethernet network>

3.1 Hardware based measurements

3.2 Software based measurements

3.3 Examples: RFC 2544 throughput and latency tests

3.4 Further reading



© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST



Content of the lesson „EtherNet/IP on Allen-Bradley platform”

1.1 Structure of the industrial control system

1.2 Integration problem



Hardware based measurements

Monitoring is carried out on the custom interface (card), and use the system for storage and formatting of the data

The important features that can be carried out on the card are:

- Timestamping
- Clock synchronization
- Dropped packet counters
- Traffic filtering

<3 Monitoring and testing the Ethernet network>
<3.1 Hardware based measurements>

Hardware based measurements

Fig. 1. Validator NT 955



Test-Um NT 955 Validator NT (Fig.1) is an all-in-one network management tool with a 4-inch color LCD screen. The NT955 Validator NT certify, identify, configure and document the Ethernet network. It defines the job, makes the tests, prints results and stores data. Test-Um NT955 Validator NT measures and presents performance results up to 1 Gigabit.

© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST



<3 Monitoring and testing the Ethernet network>
<3.1 Hardware based measurements>



Software based measurements

A software based monitor required assistance from the operating system kernel. In most operating systems the kernel contains the network stack that provides an interface between the applications and the network card. In normal operation there are two levels of filtering occurring. First the network card will only transmit to the network stack any packets that have to be dealt with by this machine. For point to point network systems, this will be all packets, for broadcast systems such as Ethernet, this will be only a subset of packets. Next the kernel splits up packets, deals with some itself, and sends on the others to the appropriate application to deal with them.

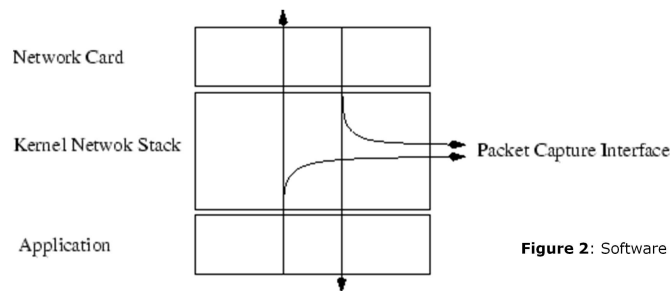


Figure 2: Software monitor setup

© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST





Example: RFC 2544 Throughput Test

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. IETF publishes recommendations, Internet standards, or network protocols as “Request for Comments”, commonly known as RFCs. RFC 2544 is an informational RFC, in other words it is *not a standard*. RFC 2544 recommendations were originally designed for performance benchmarking of network devices like routers. These recommendations have become an increasingly popular and well-accepted method to determine the performance of network links.



Example: RFC 2544 Throughput Test

Test methodology is divided into the following three tests:

1. The back-to-back test determines how the DUT responds to different quantities of frames with the minimum gap allowed by the protocol specification.
2. The frame loss test determines how the DUT responds to streams with different loading.
3. The throughput test finds the highest rate at which the DUT can forward frames.

© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST



<3 Monitoring and testing the Ethernet network>

<3.3 Examples: RFC 2544 Throughput and Latency Tests>

Example: RFC 2544 Throughput Test

Test methodology is divided into:

1. The back-to-back test with different quantities of the protocol specific
2. The frame loss test with different loading
3. The throughput test forward frames.

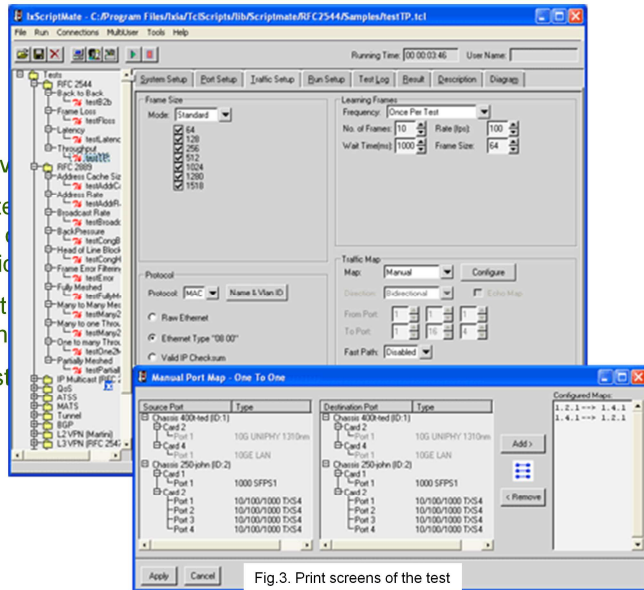


Fig.3. Print screens of the test

© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST



<3 Monitoring and testing the Ethernet network>
 <3.3 Examples: RFC 2544 Throughput and Latency Tests>

Example: RFC 2544 Throughput Test

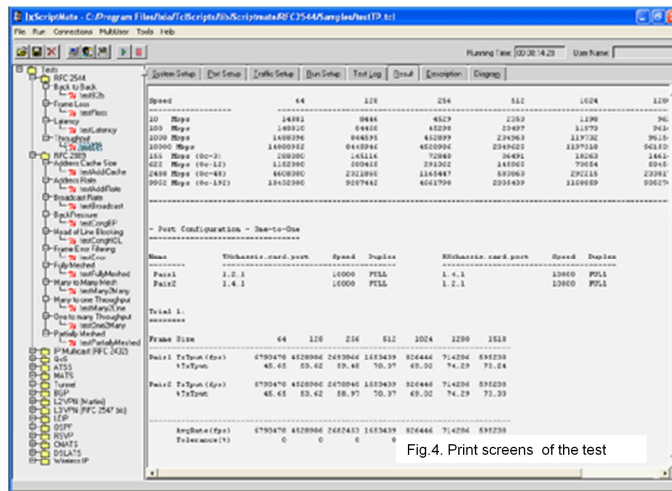


Fig.4. Print screens of the test



Example: RFC 2544 Latency Tests

To determine the latency of the DUT (*Device under test*), and how much it varies with different frame sizes. In this test, frames are transmitted for a fixed duration. Once per second, the test tags a frame and transmits it halfway through the duration time. The test compares the tagged frame's timestamp when it was transmitted with the timestamp when it was received. The difference between the two timestamps is the latency.

Example: RFC 2544 Latency Tests

To determine the latency, we need to know how much it varies from frame to frame as they are transmitted. We use the test tags a frame is transmitted over a certain duration time. The timestamp when it was received. The difference between these two is the latency.

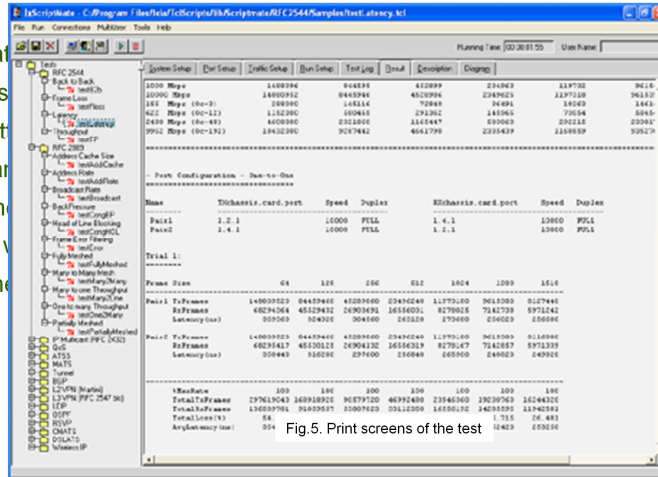


Fig.5. Print screens of the test

Example: 10GE ports handling prioritized streams

To determine the maximum rate at which the DUT can forward frames correctly according to their priority settings. The test sets up a configuration in which many ports, each with a different priority, sends traffic to one single port. The receive port may be overloaded in order to test the receipt of the highest priority frames. This test supports both MAC and IP layer frames; priority bits may be specified either in the IP precedence bits or in the 802.1p header. Latency may optionally be calculated per priority level. Test results are: the transmit and receive rates per priority, percent loss, and optionally, latency, for each priority per port.

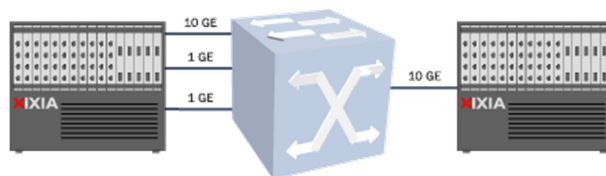


Fig. 6
Configuration for
multiports sending
traffic to single
port

© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST



<3 Monitoring and testing the Ethernet network>
<3.3 Examples: RFC 2544 Throughput and Latency Tests>

Example: How 10GE ports handle prioritized streams

To determine the maximum throughput according to their priority, many ports, each with a receive port may be overlaid with many frames. This test supports a specified either in the IP address, optionally be calculated receive rates per priority, per port.

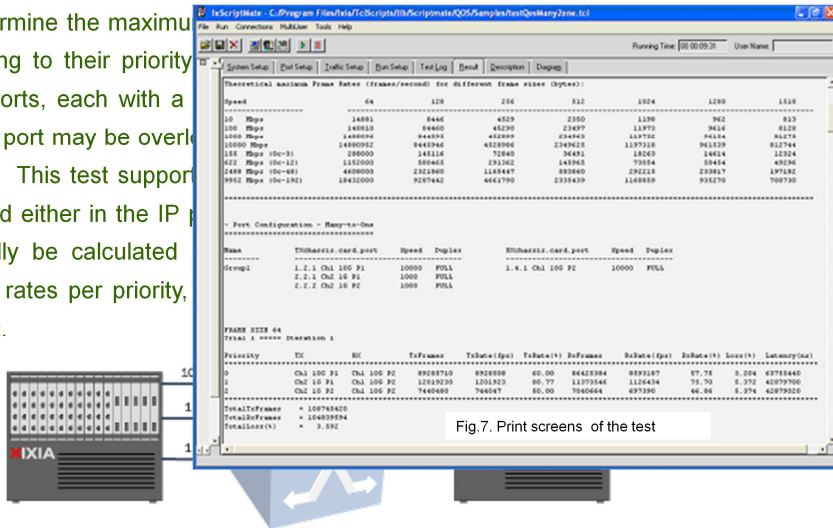


Fig.7. Print screens of the test

© 2011 P. Bogdański, W. Grega, Department of Automatics, AGH-UST



<3 Monitoring and testing the Ethernet network>
<3.3 Examples: RFC 2544 Throughput and Latency Tests>



Further reading

For more information visit:

<http://www.ixiacom.com>

<http://www.testersandtools.com>

<http://www.wand.net.nz>

<http://www.ethernetextender.com>

Sources:

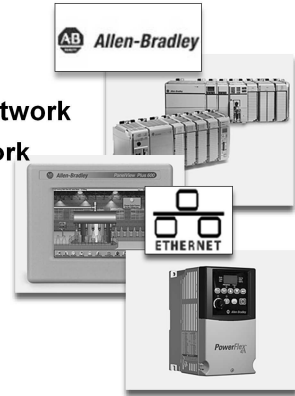
Figures (1, 2) <http://www.wand.net.nz>

Figure (6) <http://www.elnex.pl>

Figures (3, 4, 5, 7) <http://www.ixiacom.com>

CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory:
Description of laboratory and basic scenario
- 6 Software tools

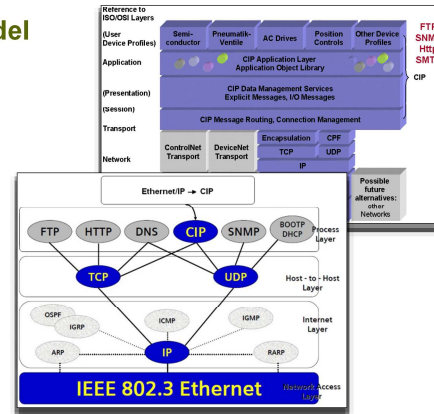


Co-operative Network Training



<4 Introduction to EtherNet/IP Technology>

- 4.1 Introduction
- 4.2 CIP and data exchange model
- 4.3 Network Topology
- 4.4 Quality of Service (QOS)
- 4.5 Further reading



© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST



Content of the lesson „EtherNet/IP on Allen-Bradley platform”

- 4.1 Introduction
- 4.2 CIP and data exchange model
- 4.3 Network Topology
- 4.4 Quality of Service (QOS)
- 4.5 Further reading



What is EtherNet/IP?

- **EtherNet/IP** is an industrial application layer protocol (CIP) operating over the Ethernet medium and used for communication between industrial control systems and their components,
- By Ethernet, we mean a TCP/UDP/IP based network
- Typically 100 MBps Twisted Pair, star topology and switch
- Could be
 - 10 MBps Coaxial
 - 1 GBps Fiber
 - 11 MBps Wireless
- By CIP we mean the *Common Industrial Protocol*. **CIP™ is an application protocol**. It defines rules for organizing and interpreting data and is essentially a messaging structure that is independent of the underlying physical layer. It is freely available and accessible to anyone, and widely supported by many manufacturers.

Spec downloadable at www.odva.org

EtherNet/IP can be easily confused as a combination of Ethernet (the physical layer, link, or medium used in most office and many industrial networking environments) and the Internet Protocol (IP)

Note that with today's technology, wireless ethernet does not successfully deliver all CIP services; we have yet to see a successful I/O implementation



What is EtherNet/IP?

EtherNet/IP :

- open industrial networking protocol,
- designed for use in process control, hard-time systems, industrial automation applications,
- wide-spread standard (low cost per node),
- classified as Class 1 Real Time Ethernet (can be extended to class 2),
- emerged from Common Industrial Protocol,
- TCP/UDP/IP encapsulation,
- ensures the desired level of service quality (QoS).

Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard. It has been developed by Rockwell Automation, managed by ODVA (Open DeviceNet Vendors Association) and designed for use in process control, hard-time systems, and industrial automation applications. (EtherNet/IP) was introduced in 2001 and today is the most developed, proven and complete industrial Ethernet network solution available for manufacturing automation. EtherNet/IP emerged due to the high demand of using the Ethernet network for control applications. Because there is wide acceptance of Ethernet technology and it is now well-established, the cost per node for Ethernet switches and other Ethernet physical media is low.

EtherNet/IP uses the tools and technologies of traditional Ethernet such as Transport Control Protocol (TCP), the Internet Protocol (IP) or User Datagram Protocol (UDP). EtherNet/IP uses of standard Ethernet TCP/IP as it is, therefore is classified as Class 1 Real Time Ethernet, according to the IEC 61 784-2. The class 1 has the largest conformity to the Ethernet TCP/IP standard and can thereby use standard hardware and software components. With the CIPsync extensions it is possible to get isochronous communication that satisfies class 2 applications. These extensions use 100 MBit/s networks with the help of IEEE 1588 time synchronisation.

EtherNet/IP was developed from a very widely implemented standard used for transferring data between two devices in DeviceNet and ControlNet called the **Common Industrial Protocol (CIP)**.

It is useful to take a look at EtherNet/IP in terms of the seven-layer Open System Interconnection (OSI) Reference Model as presented in Figure 4. 1. As with all CIP Networks, EtherNet/IP implements CIP at the Session layer and above and adapts CIP to the specific EtherNet/IP technology at the Transport. **TCP/IP encapsulation** allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP layers and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP.

History of Ethernet/IP



- ODVA founded in 1994
- Common Industrial Protocol on CAN networks
- Open network with 300+ companies providing products
- Targeted at low-end devices to provide a direct network connection
- Reduces wiring, hardware costs, start-up time and maintenance
- Increased device-level diagnostics for troubleshooting and trending
- Increased data collection from low-level devices
- Technology follows producer/consumer model

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>

www.odva.org



History

- **Mar 1998** EtherNet/IP specification for Messaging published (Part of CI Specification)
- **Dec 1999** EtherNet/IP specification for Real-time I/O published (Part of CI Specification)
- **Mar 2000** ODVA announces Support for EtherNet/IP
- **Jul 2000** First developer training for EtherNet/IP; Toolkit online
- **Nov 2000** IAONA confirms Support for EtherNet/IP: "Memorandum of Understanding" between IAONA & IDA & ODVA
- **Mar 2001** EtherNet/IP receives "Editors' Choice Award 2000" from Control Engineering
- **Apr 2001** EtherNet/IP Specification in Internet: www.odva.org or www.ethernet-ip.org or www.ethernetip.de
- **Jun 2001** ODVA publishes results of EtherNet/IP survey: More than 80 companies are designing EtherNet/IP products
- **Sep 2001** EtherNet/IP Implementor's Workshops series started at General Motors, Detroit (bi-monthly meetings)
- **Jun 2002** ODVA, IDA and PNO start collaborative WG meetings
- **Oct 2002** ODVA announces CIPsafety™
- **April 2003** ODVA announces CIPsync™

EtherNet/IP in OSI Reference Model

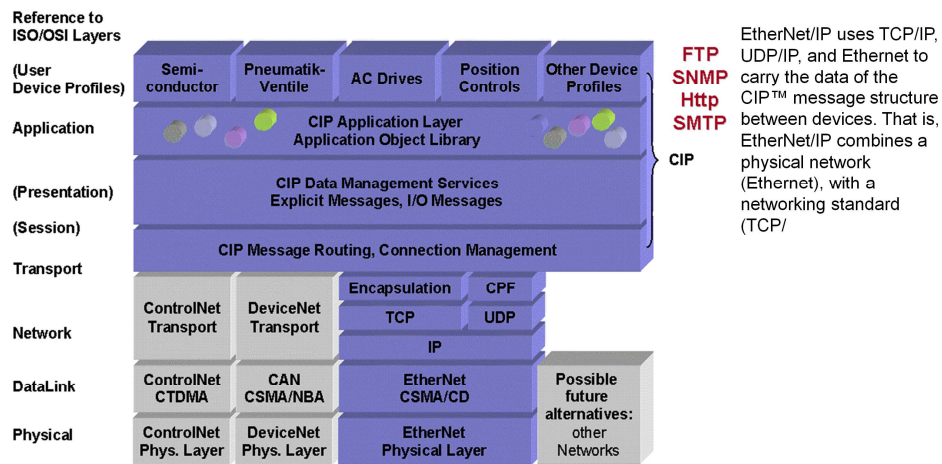


Fig. 4.1. (EtherNet/IP), Comparison of DeviceNet and ControlNet OSI [1]

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>



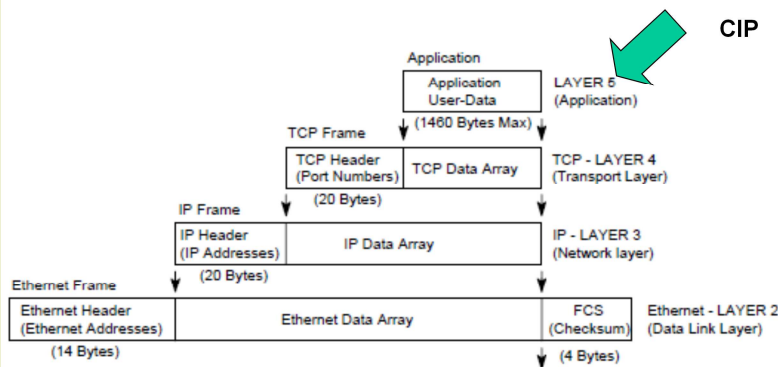
Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard. It has been developed by Rockwell Automation, managed by ODVA (Open DeviceNet Vendors Association) and designed for use in process control, hard-time systems, and industrial automation applications. (EtherNet/IP) was introduced in 2001 and today is the most developed, proven and complete industrial Ethernet network solution available for manufacturing automation. EtherNet/IP emerged due to the high demand of using the Ethernet network for control applications. Because there is wide acceptance of Ethernet technology and it is now well-established, the cost per node for Ethernet switches and other Ethernet physical media is low.

EtherNet/IP uses the tools and technologies of traditional Ethernet such as Transport Control Protocol (TCP), the Internet Protocol (IP) or User Datagram Protocol (UDP). EtherNet/IP uses of standard Ethernet TCP/IP as it is, therefore is classified as Class 1 Real Time Ethernet, according to the IEC 61 784-2. The class 1 has the largest conformity to the Ethernet TCP/IP standard and can thereby use standard hardware and software components. With the CIPsync extensions it is possible to get isochronous communication that satisfies class 2 applications. These extensions use 100 MBit/s networks with the help of IEEE 1588 time synchronisation.

EtherNet/IP was developed from a very widely implemented standard used for transferring data between two devices in DeviceNet and ControlNet called the **Common Industrial Protocol (CIP)**. It is useful to take a look at EtherNet/IP in terms of the seven-layer Open System Interconnection (OSI) Reference Model as presented in Figure 4. 1. As with all CIP Networks, EtherNet/IP implements CIP at the Session layer and above and adapts CIP to the specific EtherNet/IP technology at the Transport. TCP/IP **encapsulation** allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP layers and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP.

To obtain a desired level of **service quality** EtherNet/IP also uses **standard mechanisms** defined in layer 3 (IP) and layer 2 for Ethernet (e.g. 802.1D/Q), supported by a **proper hardware infrastructure** (e.g. multiple queue switches).

EtherNet/IP in OSI Reference Model



As you move the data down the stack of the sender, each stack layer adds its own header to the front of the message that it receives from the next higher layer. That is, the higher layers are **encapsulated** by the lower layers.

Conversely, this header information is removed by the corresponding layer at the Receiver.

Encapsulation concept

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>



Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard. It has been developed by Rockwell Automation, managed by ODVA (Open DeviceNet Vendors Association) and designed for use in process control, hard-time systems, and industrial automation applications. (EtherNet/IP) was introduced in 2001 and today is the most developed, proven and complete industrial Ethernet network solution available for manufacturing automation. EtherNet/IP emerged due to the high demand of using the Ethernet network for control applications. Because there is wide acceptance of Ethernet technology and it is now well-established, the cost per node for Ethernet switches and other Ethernet physical media is low.

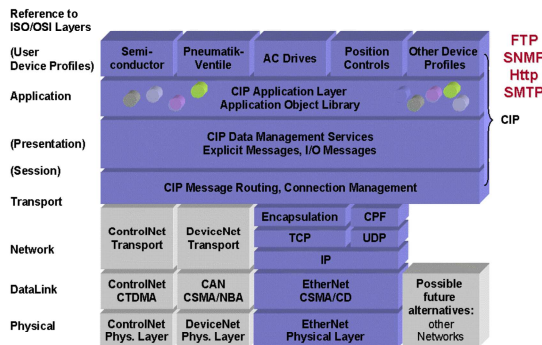
EtherNet/IP uses the tools and technologies of traditional Ethernet such as Transport Control Protocol (TCP), the Internet Protocol (IP) or User Datagram Protocol (UDP). EtherNet/IP uses of standard Ethernet TCP/IP as it is, therefore is classified as Class 1 Real Time Ethernet, according to the IEC 61 784-2. The class 1 has the largest conformity to the Ethernet TCP/IP standard and can thereby use standard hardware and software components. With the CIPsync extensions it is possible to get isochronous communication that satisfies class 2 applications. These extensions use 100 MBit/s networks with the help of IEEE 1588 time synchronisation.

Ethernet/IP was developed from a very widely implemented standard used for transferring data between two devices in DeviceNet and ControlNet called the **Common Industrial Protocol** (CIP).

It is useful to take a look at EtherNet/IP in terms of the seven-layer Open System Interconnection (OSI) Reference Model as presented in Figure 4. 1. As with all CIP Networks, EtherNet/IP implements CIP at the Session layer and above and adapts CIP to the specific EtherNet/IP technology at the Transport. TCP/IP **encapsulation** allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP layers and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP. To obtain a desired level of **service quality** EtherNet/IP also uses **standard mechanisms** defined in layer 3 (IP) and layer 2 for Ethernet (e.g. 802.1D/Q), supported by a **proper hardware infrastructure** (e.g. multiple queue switches).

EtherNet/IP in OSI Reference Model

- The Physical Layer – interaction of a single device with a medium
- The Data Link Layer – the interactions of multiple devices with a shared medium
- The Network and Transport Layers – sending messages between one or more devices



Ethernet technology by itself provides a set of physical media definitions, a scheme for sharing that physical media (CSMA/CD) and a simple frame format and source/destination addressing scheme for moving packets of data between devices on a LAN. By itself, Ethernet lacks the more complex features required of a fully-functional LAN. For that reason, all installed Ethernet networks support one or more *communications protocols* that run on top of Ethernet and provide sophisticated data transfer and network management functionality

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>



The Physical Layer

EtherNet/IP uses standard IEEE 802.3 model at the Physical and Data Link Layers. The Physical Layer as concerned primarily with the interaction of a single device with a medium. The Physical Layer is responsible for bit-level transmission between network nodes, defines specifications for electrical signals (copper networks) or the characteristics of light signals (fiber optic networks). The Physical Layer defines items such as: connector types, cable types, voltages, and pin-outs.

The Data Link Layer

The IEEE 802.3 specification is also used for transmitting packets of data from device to device on the EtherNet/IP Data Link Layer. The Data Link Layer is concerned more with the interactions of multiple devices (i.e., at least two) with a shared medium. (EtherNet/IP) uses a CSMA/CD Media Access Control (MAC) model that determines how networked devices share a common bus (i.e. cable), and how they detect and respond to packets' collisions.

The Network and Transport Layers

The Network and Transport Layers use TCP/IP Suite to send messages between one or more devices. At these layers messages used by all **CIP networks are encapsulated**. TCP/IP encapsulation allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP suite and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP.



OSI Reference Model (Network and Transport Layers)

Forms of messaging:

- Unconnected messaging
- Connected messaging (implies that the communication requires synchronization of all parties before data can be exchanged)

Network connections:

- Explicit (client-server type transactions)
- Implicit (data field contains only real-time data)

Transmission types:



EtherNet/IP uses two forms of messaging and the appropriate resources at the nodes, as defined by CIP standard (Fig.4.3)

- **Unconnected messaging** is used in the connection establishment process and for infrequent, low-priority messages. The unconnected resources in a device are referred to as the Unconnected Message Manager, or UCMM. Unconnected messages on EtherNet/IP utilize TCP/IP resources to move messages across Ethernet, asking for connection resource each time from UCMM.
- **Connected messaging** on EtherNet/IP utilizes resources within each node that are dedicated (reserved) in advance to a particular purpose, such as frequent explicit message transactions or real-time I/O data transfers. Connection resources are reserved and configured using communications services available via the CMM.

EtherNet/IP has two types of network connections: **Explicit** and **Implicit**. By using TCP/IP, (EtherNet/IP) is able to send Explicit messages, which are used to perform client-server (point to point) type transactions between nodes. For real-time messaging, EtherNet/IP uses TCP/UDP model, which allows messages to be multicast (in the sense that its target is a number of nodes in a network, and it is directed to a group of hosts/destination addresses). With Implicit messaging connection, the data field contains no protocol information, only real-time I/O data. Since the meaning of the data is pre-defined at the time the connection is established, processing time is minimized during runtime. UDP is connectionless and makes no guarantee that data will get from one device to another, however UDP messages are smaller and can be processed more quickly than TCP/IP messages. As a result, EtherNet/IP uses UDP/IP to transport I/O messages that typically contain time-critical control data.

There are three transmission types used in an (EtherNet/IP) network (Table 4.1).

Information. Non-time critical data transfers — typically large packet size. Information data exchanges are short-lived explicit connections between one originator and one target device. Information data packets use the TCP/IP protocol and take advantage of the TCP data handling features.

I/O Data. Time-critical data transfers — typically smaller packet size. I/O data exchanges are long-term implicit connections between one originator and any number of target devices. I/O data packets use the UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.

Real-time Interlocking. Cyclic data synchronization between one producer processor and any number of consumer processors. Interlocking data packets use the faster UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.



OSI Reference Model (Network and Transport Layers)

Forms of messaging:

- Unconnected messaging
- Connected messaging (synchronization of all parties before)

Network connections:

- Explicit (client-server type)
- Implicit (data field contains)

Transmission types:

ETHERNET/IP Transmission Type	Message Type	Description	Example
Information – Non time critical data transfers with typically larger packet sizes between one originator and one target	Explicit	Non-time-critical Information Data	Read/Write data by message instruction
I/O Data – Time critical data transfers with typically smaller packet sizes between one originating device and any number of target devices.	Implicit	Real-time I/O Data	Control real time data from remote I/O device
Real-Time Interlock – Synchronized cyclic data exchange between one producer and any number of consumer devices.	Implicit	Real-time Device Interlocking	Exchange real-time data between two processors

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>



EtherNet/IP uses two forms of messaging and the appropriate resources at the nodes, as defined by CIP standard (Fig.4.3)

- **Unconnected messaging** is used in the connection establishment process and for infrequent, low-priority messages. The unconnected resources in a device are referred to as the Unconnected Message Manager, or UCMM. Unconnected messages on EtherNet/IP utilize TCP/IP resources to move messages across Ethernet, asking for connection resource each time from UCMM.

- **Connected messaging** on EtherNet/IP utilizes resources within each node that are dedicated (reserved) in advance to a particular purpose, such as frequent explicit message transactions or real-time I/O data transfers. Connection resources are reserved and configured using communications services available via the CMM.

EtherNet/IP has two types of network connections connections: **Explicit** and **Implicit**. By using TCP/IP, (EtherNet/IP) is able to send Explicit messages, which are used to perform client-server (point to point) type transactions between nodes. For real-time messaging, EtherNet/IP uses TCP/UDP model, which allows messages to be multicast (in the sense that its target is a number of nodes in a network, and it is directed to a group of hosts/destination addresses). With Implicit messaging connection, the data field contains no protocol information, only real-time I/O data. Since the meaning of the data is pre-defined at the time the connection is established, processing time is minimized during runtime. UDP is connectionless and makes no guarantee that data will get from one device to another, however UDP messages are smaller and can be processed more quickly than TCP/IP messages. As a result, EtherNet/IP uses UDP/IP to transport I/O messages that typically contain time-critical control data.

There are three transmission types used in an (EtherNet/IP) network (Table 4.1).

Information. Non-time critical data transfers — typically large packet size. Information data exchanges are short-lived explicit connections between one originator and one target device. Information data packets use the TCP/IP protocol and take advantage of the TCP data handling features.

I/O Data. Time-critical data transfers — typically smaller packet size. I/O data exchanges are long-term implicit connections between one originator and any number of target devices. I/O data packets use the UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.

Real-time Interlocking. Cyclic data synchronization between one producer processor and any number of consumer processors. Interlocking data packets use the faster UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.

OSI Reference Model (Network and Transport Layers)

Forms of messaging:

- Unconnected messaging
- Connected messaging (synchronization of all parties before)

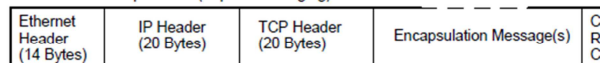
Network connections:

- Explicit (client-server type)
- Implicit (data field contains)

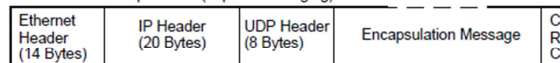
Transmission types:

ETHERNET/IP Transmission Type	Message Type	Description	Example
Information – Non time critical data transfers with typically larger packet sizes between one originator and one target	Explicit	Non-time-critical Information Data	Read/Write data by message instruction
I/O Data – Time critical data transfers with typically smaller	Implicit	Real-time I/O Data	Control real time data from remote I/O device
consumer devices.			Exchange real-time data between two processors

TCP/IP/MAC Encapsulation (Explicit Messaging)



UDP/IP/MAC Encapsulation (Implicit Messaging)



© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>



EtherNet/IP uses two forms of messaging and the appropriate resources at the nodes, as defined by CIP standard (Fig.4.3)

- **Unconnected messaging** is used in the connection establishment process and for infrequent, low-priority messages. The unconnected resources in a device are referred to as the Unconnected Message Manager, or UCMM. Unconnected messages on EtherNet/IP utilize TCP/IP resources to move messages across Ethernet, asking for connection resource each time from UCMM.
- **Connected messaging** on EtherNet/IP utilizes resources within each node that are dedicated (reserved) in advance to a particular purpose, such as frequent explicit message transactions or real-time I/O data transfers. Connection resources are reserved and configured using communications services available via the CMM.

EtherNet/IP has two types of network connections connections: **Explicit** and **Implicit**. By using TCP/IP, (EtherNet/IP) is able to send Explicit messages, which are used to perform client-server (point to point) type transactions between nodes. For real-time messaging, EtherNet/IP uses UDP/IP model, which allows messages to be multicast (in the sense that its target is a number of nodes in a network, and it is directed to a group of hosts/destination addresses). With Implicit messaging connection, the data field contains no protocol information, only real-time I/O data. Since the meaning of the data is pre-defined at the time the connection is established, processing time is minimized during runtime. UDP is connectionless and makes no guarantee that data will get from one device to another, however UDP messages are smaller and can be processed more quickly than TCP/IP messages. As a result, EtherNet/IP uses UDP/IP to transport I/O messages that typically contain time-critical control data.

There are three transmission types used in an (EtherNet/IP) network (Table 4.1).

Information. Non-time critical data transfers — typically large packet size. Information data exchanges are short-lived explicit connections between one originator and one target device. Information data packets use the TCP/IP protocol and take advantage of the TCP data handling features.

I/O Data. Time-critical data transfers — typically smaller packet size. I/O data exchanges are long-term implicit connections between one originator and any number of target devices. I/O data packets use the UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.

Real-time Interlocking. Cyclic data synchronization between one producer processor and any number of consumer processors. Interlocking data packets use the faster UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.



EtherNet/IP in OSI Reference Model

• Session, Presentation and Application Layers – CIP (Common Industrial Protocol)

© CoNeT – Co-operative Network Training

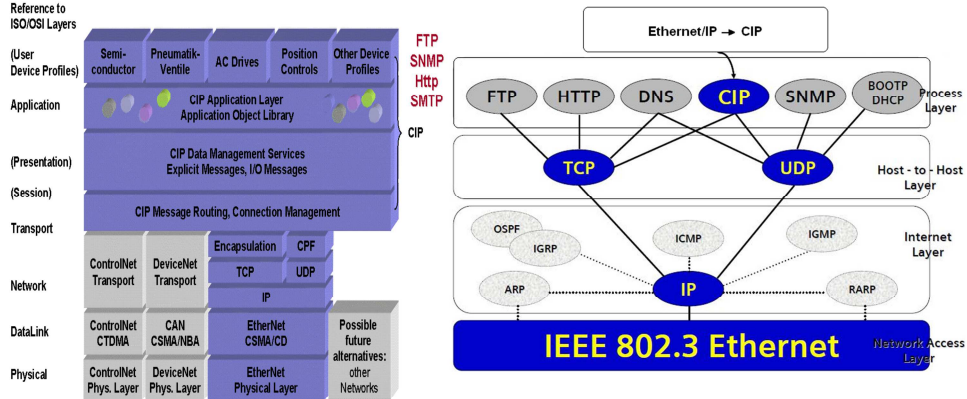


Fig. 4.2 Upper Layers with CIP protocol [1]

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST



<4 Introduction to EtherNet/IP Technology>
<4.1 Introduction>

Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard. It has been developed by Rockwell Automation, managed by ODVA (Open DeviceNet Vendors Association) and designed for use in process control, hard-time systems, and industrial automation applications. (EtherNet/IP) was introduced in 2001 and today is the most developed, proven and complete industrial Ethernet network solution available for manufacturing automation. EtherNet/IP emerged due to the high demand of using the Ethernet network for control applications. Because there is wide acceptance of Ethernet technology and it is now well-established, the cost per node for Ethernet switches and other Ethernet physical media is low.

EtherNet/IP uses the tools and technologies of traditional Ethernet such as Transport Control Protocol (TCP), the Internet Protocol (IP) or User Datagram Protocol (UDP). EtherNet/IP uses of standard Ethernet TCP/IP as it is, therefore is classified as Class 1 Real Time Ethernet, according to the IEC 61 784-2. The class 1 has the largest conformity to the Ethernet TCP/IP standard and can thereby use standard hardware and software components. With the CIPsync extensions it is possible to get isochronous communication that satisfies class 2 applications. These extensions use 100 MBit/s networks with the help of IEEE 1588 time synchronisation.

EtherNet/IP was developed from a very widely implemented standard used for transferring data between two devices in DeviceNet and ControlNet called the **Common Industrial Protocol (CIP)**.

It is useful to take a look at EtherNet/IP in terms of the seven-layer Open System Interconnection (OSI) Reference Model as presented in Figure 4. 1. As with all CIP Networks, EtherNet/IP implements CIP at the Session layer and above and adapts CIP to the specific EtherNet/IP technology at the Transport. TCP/IP **encapsulation** allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP layers and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP. To obtain a desired level of **service quality** EtherNet/IP also uses **standard mechanisms** defined in layer 3 (IP) and layer 2 for Ethernet (e.g. 802.1D/Q), supported by a **proper hardware infrastructure** (e.g. multiple queue switches).



CIP Protocol

Every network device – a series of objects

3 types of objects:

- Required objects (e.g. Identity object, Message router object, Network object)
- Application objects
- Vendor Specific objects
- Access to the device requires object number, instance number and attribute number

The type of messaging required will determine which specific transport layer protocol will be used, TCP (explicit/information), or UDP (implicit/control).

Fig. 4.3. A typical CIP device representation [1]

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
 <4.2 CIP and data exchange model>



In the CIP Protocol, every network device represents itself as a series of objects. Each object is simply a grouping of the related data values in a device. There are three types of objects defined by the CIP protocol and used by Ethernet/IP data representation.

Required Objects

Objects required by the specification of every CIP device. For example:

Identity object: contains identity data called **attributes** (ex. Vendor ID, Date of manufacturer, Device serial number and other identity data)

Message Router object: this object routes explicit request messages between objects in a device

Network object: contains the physical connection data for the object such as IP address and other data describing the interface to the Ethernet port on the device

Application Objects

Application Objects allow the user to organize the data that are specific to a particular kind of device. These objects define the data encapsulated by the device. They are specific to the device type and function. For example an analog I/O device can be described in object term by attributes such as: type, resolution, values of input and output.

These application layer objects are predefined for a large number of common device types. The same type of CIP devices must contain the same series of application objects. The series of application objects for a particular device type is known as the device profile.

Vendor Specific Objects

Objects not found in the profile for a device class are termed Vendor Specific. These objects are included by the vendor as additional features of the device. The CIP protocol provides access to these vendor extension objects in exactly the same way as either application or required objects.

Accessing regular pieces of data from the network to the device requires:

- Object Number
- Instance Number (Instances are the way of organizing the same kind of data , e.g., sharing same attributes)
- Attribute Number

A typical CIP device is shown in Fig. 4.3

CIP Protocol

In the Object Model shown access to the internal object model of any device is controlled by one of 2 objects, the **Unconnected Message Manager** and the **Connection Manager**. This is because EtherNet/IP is a connection-based network. A CIP connection defines a packet that will be produced on the network.

When a connection is established, the transmissions associated with that connections are assigned a Connection ID (CID). If the connection involves a bidirectional exchange, then two Connection ID values are assigned. Since most messaging on a CIP network is done through connections, a process has been defined to establish such connections between devices that are not "connected" yet.

This is done through the Unconnected Message Manager (UCMM), which is responsible for processing the connection requests. Once a connection has been established, then all communication resources needed in the devices, including any intermediate CIP bridges/routers are reserved. The overall network loading and bandwidth required for that data exchange is minimized.

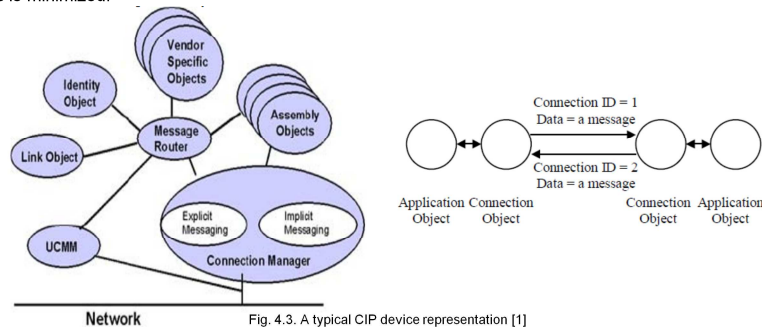


Fig. 4.3. A typical CIP device representation [1]

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.2 CIP and data exchange model>



In the CIP Protocol, every network device represents itself as a series of objects. Each object is simply a grouping of the related data values in a device. There are three types of objects defined by the CIP protocol and used by Ethernet/IP data representation.

Required Objects

Objects required by the specification of every CIP device. For example:

Identity object: contains identity data called **attributes** (ex. Vendor ID, Date of manufacturer, Device serial number and other identity data)

Message Router object: this object routes explicit request messages between objects in a device

Network object: contains the physical connection data for the object such as IP address and other data describing the interface to the Ethernet port on the device

Application Objects

Application Objects allow the user to organize the data that are specific to a particular kind of device. These objects define the data encapsulated by the device. They are specific to the device type and function. For example an analog I/O device can be described in object term by attributes such as: type, resolution, values of input and output.

These application layer objects are predefined for a large number of common device types. The same type of CIP devices must contain the same series of application objects. The series of application objects for a particular device type is known as the device profile.

Vendor Specific Objects

Objects not found in the profile for a device class are termed Vendor Specific. These objects are included by the vendor as additional features of the device. The CIP protocol provides access to these vendor extension objects in exactly the same way as either application or required objects.

Accessing regular pieces of data from the network to the device requires:

- Object Number
- Instance Number (Instances are the way of organizing the same kind of data , e.g., sharing same attributes)
- Attribute Number

A typical CIP device is shown in Fig. 4.3

Data Exchange Model

Producer-Consumer data exchange model – how data is exchanged between applications in devices

- Input Module (producer) – Controller (consumer),
- Controller can be also a producer,
- Message identified by its connection ID,
- Request Packet Interval (RPI)

Forms of messaging:

- Unconnected messaging
- Connected messaging

When a message is introduced into the network it is identified by its **connection ID** not by its destination address.

Multiple nodes may then consume the data to which the connection ID refers. As a result, when a node wants to receive data it only needs to ask for it once in order to consume the data each time it is produced.

If subsequent nodes want to receive the same data simultaneously all they need to know is the connection ID. As a result we get much more efficient use of bandwidth.

When one adds a module to the I/O configuration of a controller, the **Requested Packet Interval (RPI)** must be entered as a parameter. This value specifies how often to produce the data for that device. For example, if one specifies an RPI of 50 ms, it means that every 50 ms the device should send its data to the controller or the controller should send its data to the device.

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.2 CIP and data exchange model>



Ethernet/IP uses **the producer-consumer** data exchange model which describes rules for how data is exchanged between application programs running in devices.

The CIP producer/consumer networking model replaces the old source/destination (master/slave) model. In traditional I/O systems, controllers **poll** input modules to obtain their input status. In the CIP system, input modules are not polled by a controller. Instead, they produce (multicast) their data either upon a change of state or periodically. With implicit connections, messages are sent cyclically. The frequency of update depends upon the options chosen during configuration.

The input module, therefore, is a producer of input data, and the controller is a consumer of the data. The controller can also produce data for other controllers to consume. Information generated by one device can be consumed a group of devices over the EtherNet/IP network.

When a message is introduced into the network it is identified by its connection ID not by its destination address. Multiple nodes may then consume the data to which the connection ID refers. As a result, when a node wants to receive data it only needs to ask for it once in order to consume the data each time it is produced. If subsequent nodes want to receive the same data simultaneously all they need to know is the connection ID. As a result we get much more efficient use of bandwidth.

When one adds a moduler to the I/O configuration of a controller, must enter the **Requested Packet Interval (RPI)** as a parameter. This value specifies how often to produce the data for that device. For example, if one specifies an RPI of 50 ms, it means that every 50 ms the device should send its data to the controller or the controller should send its data to the device.

The following table (4.2) categorizes the message Transport Classes supported by EtherNet/IP.

Data Exchange Model

Recall that **explicit message connections are point-to-point communication** paths between two devices. They follow a simple request/response network communication format and are always made to the message router (the Message Router Object). Each request contains explicit information (not time critical) that the receiving node decodes and acts upon, then generates an appropriate response.

Thus, all explicit connections are direct connections between two devices, which require a source address, a destination address, and a connection ID in each direction. Explicit messages are normally triggered by events that are external to the CIP™ application layer.

Forms of messaging:

- Unconnected messaging
- Connected messaging

Implicit message connections provide dedicated special purpose communication paths (or ports) between a producer application object and one or more consumer application objects. They follow the producer/consumer-based connection model and contain implicit (timesensitive data).

The data is implicit because it is identified at the time that the connection is established and the connection ID's are assigned and we say that the data is implicitly defined by its connection ID. Implicit messaging is commonly used for I/O messages and takes place within the application layer of the protocol with both the producer node and consuming nodes aware of the message content before transmission. For implicit communication, the UDP/IP/MAC protocol stack is used, which supports the **multicast communication** (UDP also supports unicast and broadcast communication, while TCP is restricted to unicast only).

UDP packets are not transmitted directly to the actual IP address of a receiving device, but are transmitted using a specific device allocated IP multicast address.

Table 4.2 Traffic classes

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.2 CIP and data exchange model>



Ethernet/IP uses **the producer-consumer** data exchange model which describes rules for how data is exchanged between application programs running in devices.

The CIP producer/consumer networking model replaces the old source/destination (master/slave) model. In traditional I/O systems, controllers **poll** input modules to obtain their input status. In the CIP system, input modules are not polled by a controller. Instead, they produce (multicast) their data either upon a change of state or periodically. With implicit connections, messages are sent cyclically. The frequency of update depends upon the options chosen during configuration.

The input module, therefore, is a producer of input data, and the controller is a consumer of the data. The controller can also produce data for other controllers to consume. Information generated by one device can be consumed a group of devices over the EtherNet/IP network.

When a message is introduced into the network it is identified by its connection ID not by its destination address. Multiple nodes may then consume the data to which the connection ID refers. As a result, when a node wants to receive data it only needs to ask for it once in order to consume the data each time it is produced. If subsequent nodes want to receive the same data simultaneously all they need to know is the connection ID. As a result we get much more efficient use of bandwidth.

When one adds a moduler to the I/O configuration of a controller, must enter the **Requested Packet Interval (RPI)** as a parameter. This value specifies how often to produce the data for that device. For example, if one specifies an RPI of 50 ms, it means that every 50 ms the device should send its data to the controller or the controller should send its data to the device.

The following table (4.2) categorizes the message Transport Classes supported by EtherNet/IP.



Network Topology

Network Devices – Hubs, Switches, Routers, Repeaters, standard twisted-pair and fiber-optic cables.

Active Star topology – simple to wire, easy to debug and maintain connections

DETERMINISM can be ensured by:

- Replacement of the typical hubs by intelligent switches
- Switching to higher baud rate
- Full duplex implementation

Table 4.3 Half-duplex v. full duplex ETHERNET

Half-duplex Ethernet	Full-duplex Ethernet
<ul style="list-style-type: none"> • Can work in broadcast mode • Mandatory to support • When two devices try to transmit at the same time a collision occurs • Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol resolves collisions, but generates delays 	<ul style="list-style-type: none"> • Restricted to point to point links connecting exactly two stations • Requires switching • Each side sends with no blocking on anything • Receiving is independent of sending • Quality of service better than half-duplex • The preferred real-time mode

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST

<4 Introduction to EtherNet/IP Technology>
<4.3 Network Topology>



When building a network, users may use many of the following components: cabling, transceivers, hubs, repeaters, routers, and switches. Standard twisted-pair and fiber-optic cables are fully functional with EtherNet/IP. Depending on the network configuration, an Ethernet hub or switch is appropriate.

Hub is an inexpensive connectivity method that provides an easy method of connecting devices on information networks (shared Ethernet).

Switch reduces collisions and is recommended for real-time control installations (switched Ethernet).

Routers are used to isolate control data traffic from other types of office data traffic, to isolate information traffic on the plant floor from control traffic on the plant floor, and for security purposes, i.e., firewalls.

Repeaters extend the overall network cable length. They can also connect networks with different media types.

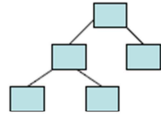
To successfully apply EtherNet/IP in automation, the issue of determinism has to be considered. The inherent principle of the Ethernet bus access mechanism – whereby collisions are detected and resolved using CSMA/CD protocol – cannot guarantee determinism.

First of all, the typical hubs used in an office environment have to be replaced by intelligent switches that will forward only those Ethernet frames that are intended for nodes connected to this switch. With the use of switch technology, collisions are largely avoided except for those cases where two or more messages are sent to the same node at the same time. The situation can be further improved by switching to a higher baud rate without increasing the number of messages. This will result in a lower bandwidth utilization and will thus reduce the chance of collision.

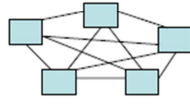
Typically an EtherNet/IP network uses an active star topology where groups of devices are connected point-to-point to a switch. The benefit of a star topology is in its support of both 10 and 100M bit/s products. Mixing 10 and 100M bit/s is possible, and most Ethernet switches will negotiate the speed automatically. The star topology offers connections that are simple to wire, easy to debug and easy to maintain. The use of Ethernet controller chips, wiring and switches that support **full duplex** operation eliminates collisions on the network and permits a node to simultaneously transmit and receive messages effectively (Table 4.3). Implemented full duplex increases the level of determinism.



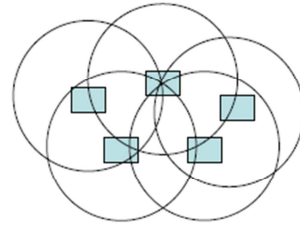
Network Topology



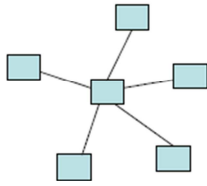
Tree



Mesh (wired)



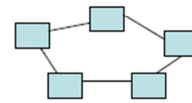
Mesh (wireless)



Star



Bus



Ring



Quality of Service (QoS) improvement

QoS principles:

- Distinguishing one traffic stream from another (classification)
- Assigning a label to each kind of traffic
- Providing different treatment to different traffic classes

Several different QoS mechanisms have been defined for network protocols. For EtherNet/IP one of the examples is IEEE's 802.1D/Q standard.

EtherNet/IP specification ed. 1.6 defines the behavior of EtherNet/IP networks with respect to QoS. The overall approach calls for devices to mark their packets with a priority value, using IEEE 802.1D/Q standard or DiffServ Code Points. Switches and routes must be able to differentiate real-time traffic from non-critical traffic streams (e.g. implicit vs. explicit messages).

Tab. 4.4 Message Categories vs QoS

QoS Class	Application	QoS Latency	Jitter
1	Controller-to-controller	100 ms	-
2	Distributed I/O devices	< 25% packet interval	Up to some maximum tolerable
3	Motion control	≥1ms	≥1μs

© 2011 Wojciech Modzelewski, Wojciech Grega, Department of Automatics, AGH-UST



<4 Introduction to EtherNet/IP Technology>
<4.4 Quality of Service (QoS)>

When building a control network, it is important to understand the relative importance of the various quality of service (QoS) measurements have on the various types of messages.

Summaries are provided in the Table 4.4. The definition of these rough classes is based on the experiences with existing classes of applications in the bus technologies.

The following QoS principles are employed in the networks:

- Distinguishing one traffic stream from another (classification)
- Assigning a label to each kind of traffic
- Providing different treatment to different traffic classes

The EtherNet/IP specification ed. 1.6 defines behavior of EtherNet/IP networks with respect to QoS. The overall approach calls for devices to mark their packets with a priority value, using IEEE 802.1D/Q standard or Differserv Code Points. Switches and routes must be able to differentiate real-time traffic from non-critical traffic streams (e.g. implicit vs. explicit messages).

Several different QoS mechanisms have been defined for network protocols. For EtherNet/IP one of the examples is IEEE's 802.1D/Q standard. Using priorities according to IEEE 802.1D/Q creates an effective method for isolating time-critical and best-effort data.

IEEE 802.1D defines the use of priority in the IEEE 802.1Q format, while the IEEE's 802.1Q standard was developed to address the problem of how to break large networks into smaller parts so broadcast and multicast traffic wouldn't grab more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks. The 802.1Q specification establishes a standard method for inserting virtual LAN (VLAN) membership information into Ethernet frames.

The general QoS approach for EtherNet/IP calls for devices to mark their paskets with a priority value. By this marking switches are able to differentiate EtherNet/IP traffic from non-real time traffic as well as to differentiate I/O Data from Explicite Messages.

For example, the QoS behavior of EtherNet/IP is supported by the following solutions:

- For CIP transport class 0 and 1 UDP connections, there is defined mapping of CIP priorities to 802.1D priorities (Layer 2) or other standard codes of differentiated services (Layers 3:RFC 2475)
- For CIP transport class 3 connections (TCP), there is a single defined 802.1D priority value



Quality of Service (QoS)

The general QoS approach to EtherNet/IP calls for devices to mark their packets with a priority value. From this marking, switches are able to differentiate EtherNet/IP traffic from non-real time traffic as well as differentiate I/O Data from Explicit Messages.

For example, the QoS behavior of EtherNet/IP is supported by the following solutions:

- For CIP transport class 0 and 1 UDP connections, there is a defined mapping of CIP priorities to 802.1D priorities (Layer 2) or other standard codes of differentiated services (Layers 3:RFC 2475)
- For CIP transport class 3 connections (TCP), there is a single defined 802.1D priority value



When building a control network, it is important to understand the relative importance of the various quality of service (QoS) measurements have on the various types of messages.

Summaries are provided in the Table 4.4. The definition of these rough classes is based on the experiences with existing classes of applications in the bus technologies.

The following QoS principles are employed in the networks:

- Distinguishing one traffic stream from another (classification)
- Assigning a label to each kind of traffic
- Providing different treatment to different traffic classes

The EtherNet/IP specification ed. 1.6 defines behavior of EtherNet/IP networks with respect to QoS. The overall approach calls for devices to mark their packets with a priority value, using IEEE 802.1D/Q standard or Differserv Code Points. Switches and routes must be able to differentiate real-time traffic from non-critical traffic streams (e.g. implicit vs. explicit messages).

Several different QoS mechanisms have been defined for network protocols. For EtherNet/IP one of the examples is IEEE's 802.1D/Q standard. Using priorities according to IEEE 802.1D/Q creates an effective method for isolating time-critical and best-effort data.

IEEE 802.1D defines the use of priority in the IEEE 802.1Q format, while the IEEE's 802.1Q standard was developed to address the problem of how to break large networks into smaller parts so broadcast and multicast traffic wouldn't grab more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks. The 802.1Q specification establishes a standard method for inserting virtual LAN (VLAN) membership information into Ethernet frames.

The general QoS approach for EtherNet/IP calls for devices to mark their packets with a priority value. By this marking switches are able to differentiate EtherNet/IP traffic from non-real time traffic as well as to differentiate I/O Data from Explicit Messages.

For example, the QoS behavior of EtherNet/IP is supported by the following solutions:

- For CIP transport class 0 and 1 UDP connections, there is defined mapping of CIP



Further Reading

For more information visit:

<http://www.odva.org>
<http://www.rtaautomation.com>
<http://en.wikipedia.org/wiki/EtherNet/IP>
<http://www.cisco.com>

Figure and table sources:

[1] - <http://www.ethernetip.de> Industrial Ethernet Symposium, Amsterdam 2005
[2] <http://www.ethernetip.de/> Developer Guide

For more information visit:

<http://www.odva.org>
<http://www.rtaautomation.com>
<http://en.wikipedia.org/wiki/EtherNet/IP>
<http://www.cisco.com>

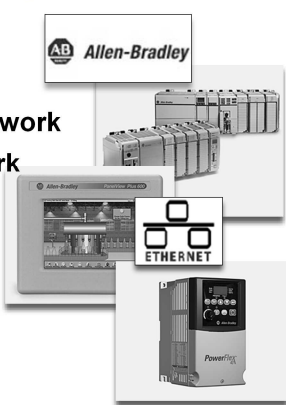
Figure and table sources:

[1] - <http://www.ethernetip.de> Industrial Ethernet Symposium, Amsterdam 2005
[2] <http://www.ethernetip.de/> Developer Guide

Lesson „EtherNet/IP on Allen-Bradley platform”


CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory:
Description of laboratory and basic scenario**
- 6 Software tools



© CoNeT – Co-operative Network Training

Co-operative Network Training
Education and Culture DG
Lifelong Learning Programme



CoNet Mobile Lab: Ethernet IP on Allen Bradley platform

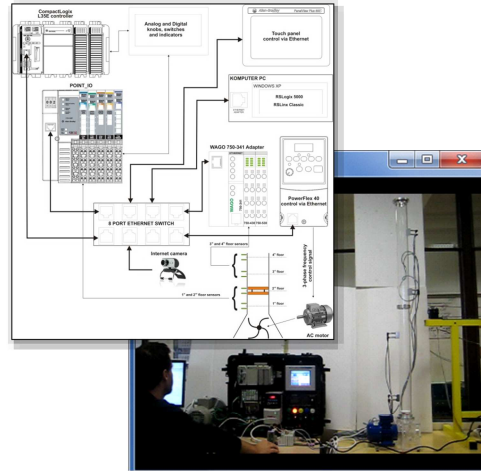
Introduction

- 1 Distributed Control Architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory: Description of laboratory and basic scenario
- 6 Software tools

© 2011 Maciej Rosól, Adam Piłat, Department of Automatics, AGH University of Science and Technology

<5 Introduction to laboratory: Description of laboratory and basic scenario>

- 5.1 System architecture
- 5.2 Aerolift overview
- 5.3 Basis of PLC
- 5.4 Real-time network structure
- 5.5 References



© 2011 Maciej Rosól, Adam Pilat, Department of Automatics, AGH-UST



Content of the lesson „EtherNet/IP on Allen-Bradley platform“

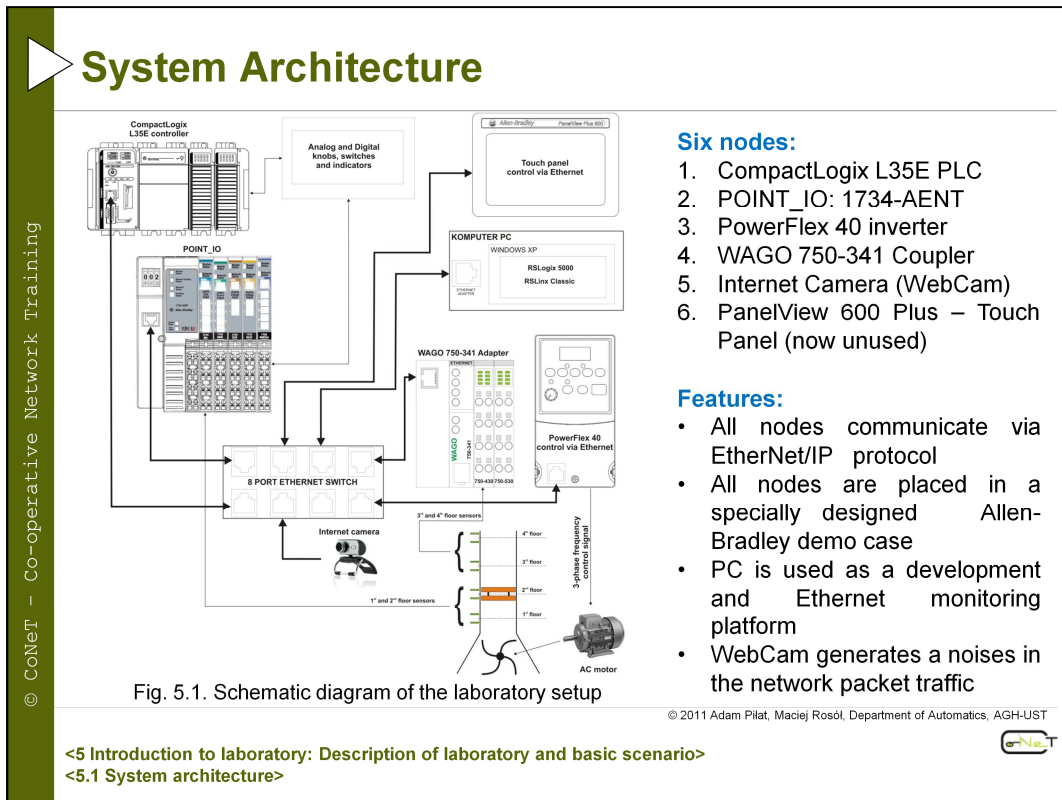
5.1 System architecture

5.2 Aerolift Overview

5.3 Basis of PLC

5.4 Real-time network structure

5.5 References



The structure of the Ethernet/IP network is depicted in Fig. 5.1. It contains five nodes, Aero Lift process control, computer and analog/digital controls (knobs, switches) and indicators. The main node is the CompactLogix L35E PLC controller made by Allen&Bradley company. The other nodes are: POINT_IO, WAGO 750-341 I/O adapter, PowerFlex 40E inverter and PanelView Plus600 touch panel. The all nodes are connected via Ethernet media channel with Ethernet/IP industrial protocol. The host computer is used to configure the all elements of the network, program the PLC controller and analyzing the Ethernet/IP packets transmitted over TCP/IP or UDP protocol. An internet camera (Webcam) is used to generate additional packet traffic on the Ethernet network.

Each of the nodes in the network provides a following function:

- CompactLogix L35E PLC controller executing a control algorithms. It receives the Ethernet packet containing the measurement data from the POINT_IO and WAGO 750-341 I/O adapter, than produces control value and sends it (in the form of Ethernet packet) to the PowerFlex 40E inverter.
- POINT_IO and WAGO 750-341 I/O adapter are directly connected to the Aero Lift process. They measure the position of the movable lift by utilizing a digital capacitance sensors. The signal from the sensors are connected to digital input modules.
- PowerFlex 40E inverter provides the required power and adjusts a speed for AC motor. The reference rotational speed signal is received as a Ethernet packet from the CompactLogix L35E PLC controller.
- PanelView Plus600 touch panel is not normally used in the project. However it can be used as an HMI (Human Machine Interface). Connection to a PLC controller can be done using either RSLinx or KEP OPC Server software.

The all nodes of the Ethernet/IP network was placed in a specially designed Allen-Bradley demo case.



Compact Logix Controller 1769-L35E



PLC CLC 1769-L35E is equipped with:

- CPU (Central Processing Unit)– firmware revision 16.3, 15 MB internal memory,
- one RS232 serial port and one 100Mb/s EtherNet/IP port,
- the Compact Flash card socket,
- the power supply Allen-Bradley 1769-PA2: input:120/240VAC, output: 24VDC,
- the Digital I/O module Allen-Bradley 1769-IQ6XOW4 (firmware revision 2.1 series B),
- the analog I/O module Allen-Bradley 1769-IF4XOF2 (firmware revision 1.1 series A),
- the terminal of the CompactBUS Allen-Bradley 1769-ERC.



The main part of the Allen-Bradley demo case is Compact Logix Controller 1769-L35E. The controller consist of the following elements:

CPU (Central Processing Unit)– firmware revision 16.3,

- 15 MB internal memory,
- one RS232 serial port and one 100Mb/s EtherNet/IP port,
- the Compact Flash card socket,
- the power supply Allen-Bradley 1769-PA2, input :120/240VAC, output: 24VDC,
- the Digital I/O module Allen-Bradley 1769-IQ6XOW4 (firmware revision 2.1 series B),
- the analog I/O module Allen-Bradley 1769-IF4XOF2 (firmware revision 1.1 series A),
- the terminal of the CompactBUS Allen-Bradley 1769-ERC.

Compact Logix Controller 1769-L35E - local I/O Modules

Table 5.1. The main parameters of the CompactLogix L35E local I/O modules

1756-ENBT	1769-IQ6XOW4	1769-IF4XOF2
<ul style="list-style-type: none"> Interface for a ControlLogix controller to communicate with other devices over an EtherNet/IP network, Adapter for 1756 I/O modules, Web server to provide diagnostic and status information, Communication via produced/consumed tags and MSG instructions. 	<ul style="list-style-type: none"> 6 digital inputs 24V DC (sinking/sourcing), operating voltage range 10 to 30 V, 4 digital outputs 24V relay (AC/DC), operating voltage range 5 to 265V AC and 5 to 125V DC I/O diagnostic LEDs. 	<ul style="list-style-type: none"> 4 analog inputs (differential or single-ended), analog normal operating ranges: voltage 0-10V, current 0-20 mA, resolution: 8-bits plus sign, response time: 5 ms/channel, 2 analog outputs (single-ended), analog normal operating ranges: voltage 0-10V, current 0-20 mA, resolution: 8-bits plus sign, response speed: 0.3 (resistance, inductor), 3 ms (capacitance).

© 2011 Adam Pilat, Maciej Rosol, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.1 System architecture>



The main parameters of the CompactLogix L35E local I/O modules are presented in Table 5.1. The 1769-L35E controller is designed for the mid-range application. It is equipped in the operating system with preemptive multitasking system. This environment supports as many as 8 tasks, but only one can be continuous. A task can have as many as 32 separate programs with own executable routines and program tags.

POINT_IO Modules

Table 5.2. Parameters of the distributed POINT_IO modules

1734-AENT	<ul style="list-style-type: none"> • Serves as a bridge between POINT I/O modules and the Ethernet/IP network, • Provides communication for <i>CompactLogix</i>, <i>ControlLogix</i> controllers (supports of connections from multiple controllers simultaneously), • Communication via produced/consumed tags, • EtherNet/IP messages encapsulated within standard TCP/UDP/IP protocol, • Half/full duplex 10 Mbit or 100 Mbit operation (RJ-45, interfacing via category 5 rated twisted pair cable).
1734-IB8	<ul style="list-style-type: none"> • 8 digital inputs module: 24 V DC, sink, • Operating voltage range: 10...28.8 V DC. • Allows input filter time in the range of 0...63 ms.
1734-OB4E	<ul style="list-style-type: none"> • 4 digital outputs module: 24 V DC, source, • Output current rating max. 1 A/channel, • Outputs are not isolated, • Operating voltage range: 10...28.8 V DC.



The 1734-AENT adapter can carry up to six I/O modules. They are mounted in a POINT_IO's chassis and communicate with 1734-AENT adapter via internal bus. Basic parameters of the 1734-AENT and installed I/O modules are summarized in Table 5.2.

POINT_IO Modules

Table 5.2. Parameters of the distributed POINT_IO modules

1734-IE2V	<ul style="list-style-type: none"> • 2 analog inputs module. Operating ranges voltage: -10... +10 V. • Input resolution: 15-bits plus sign (-32,768...+32,767), • The module produces 6 bytes of input data and fault status data: 2-bytes data/channel, 1-byte status/channel, • Operates in unipolar or bipolar mode.
1734-OE2V	<ul style="list-style-type: none"> • 2 analog outputs module. Output voltage signal range: 0... +10 V or -10... +10 V, • Output resolution: 13-bits plus sign (-32,768...+32,767), • The module consumes 4 bytes of output data: 2-bytes/channel, • The module produces 2 bytes of fault status data: 1-byte/channel, • Operates in unipolar or bipolar mode.
1734-VHSC24	<ul style="list-style-type: none"> • Very High Speed Counter module: 24V, • Accepts feedback from an encoder (either single ended or differential), pulse generators, or mechanical limit switches at frequencies up to 1 MHz, • Allows filtering with four settings (50Hz, 500Hz, 5kHz or 50kHz).



© CoNeT – Co-operative Network Training

© 2011 Adam Pilat, Maciej Rosól, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.1 System architecture>



▶ WAGO I/O Modules

Table 5.3. The I/O modules of the WAGO node

750-430	<ul style="list-style-type: none"> • 8-channel digital input module, DC 24 V, • Each input module has an RC noise rejection filter with a time constant of 3.0 ms, • The status of the input channels is indicated via status LEDs, • 1-conductor connection, high-side switching allowing direct connection to pnp-type digital sensors, • The process image of the module occupies 1-byte.
750-530	<ul style="list-style-type: none"> • 8-channel digital output module, DC 24 V 0.5 A, • The status of the input channels is indicated via status LEDs, • Short-circuit-protected, high-side switching, • The process image of the module occupies 1-byte.



© 2011 Adam Pilat, Maciej Rosol, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.1 System architecture>



The fieldbus coupler 750-341 accepts the all peripheral I/O modules in the WAGO-I/O-SYSTEM 750. When power is applied to the fieldbus coupler, it automatically detects all I/O modules connected to the coupler and creates a local process image. This can be a mixture of analog and digital modules. The process image is subdivided into an input and an output data area. The 750-341 coupler is able to send/receive process data via Ethernet and supports a series of network protocols. For the exchange of process data, the MODBUS TCP (UDP) and the Ethernet/IP protocols are available. However, the two communication protocols cannot be used together. The other protocols like HTTP, BootP, DHCP, DNS, SNTP, FTP and SNMP are provided for the management and diagnosis of the system. The main parameters of the installed digital I/O modules are shown in Table 5.3.

▶ Allen-Bradley PowerFlex 40 AC

Main features of the PowerFlex40 AC drive:

- Integral keypad for simple operation and programming,
- 4 digit display with 10 LED indicators for display of drive status,
- Communication with PC using the RS-485 interface ,Ethernet/IP (also DeviceNet, PROFIBUS DP, LonWorks and ControlNet interface are available),
- Autotune allows to adapt to individual motor characteristics,
- Sensorless Vector Control provides exceptional speed regulation and very high levels of torque across the entire speed range of the drive,
- Built-in PID controller,
- Timer, Counter, Basic Logic and StepLogic functions,
- Built-in digital and analog I/O (2 analog inputs, 7 digital inputs (4 fully programmable), 1 analog output, 3 digital output),
- Easy set-up over the network (RS NetWorx property).



<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.1 System architecture>



Allen-Bradley PowerFlex 40 AC drive is the smallest and most cost-effective members of PowerFlex family of drives. The PowerFlex 40 is designed to use for speed control in applications such as machine tools, fans, pumps and conveyors and material handling systems. Main features of the PowerFlex40 AC drive:

- integral keypad for simple operation and programming,
- 4 digit display with 10 LED indicators for display of drive status,
- communication with PC using the RS-485 interface, Ethernet/IP (also DeviceNet, PROFIBUS DP, LonWorks and ControlNet interface are available),
- Autotune allows to adapt to individual motor characteristics,
- Sensorless Vector Control provides exceptional speed regulation and very high levels of torque across the entire speed range of the drive,
- built-in PID controller,
- Timer, Counter, Basic Logic and StepLogic functions,
- built-in digital and analog I/O (2 analog inputs, 7 digital inputs (4 fully programmable), 1 analog output, 3 digital output),
- easy set-up over the network (RS NetWorx property).

▶ PanelView Plus 600 & EtherNet/IP Configuration



- Works as an operator interface,
- Works under Windows CE operating system,
- Communication via Ethernet interface,
- Has possibilities in data presenting, trends and data collection,
- Visualization can be implemented using RSView Studio environment.

The information required to configure the network:

- Parameters of the Ethernet network: IP address, Subnet mask, Gateway address.
- Types and parameters of the installed modules.
- Requested Packet Interval (RPI) time.

The **RPI** is a common parameter configuration for all the modules connected to a network. It specifies the period at which data is updated over a connection.

© 2011 Adam Pilat, Maciej Rosól, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.1 System architecture>



The PanelView Plus 600 is an operator interface. It is equipped with a 5.5 inch display with touch screen. It works under Windows CE. The panel has wide possibilities in presenting data such as animations, trends and data collection. The visualization can be implemented using RSView Studio environment. Communication with the panel is using the Ethernet interface. The data exchange between Ethernet/IP devices and PanelView uses OPC client/server mechanism.

EtherNet/IP Configuration

Configuration is performed using RSLinx and RSLogix 5000 software.

Table 5.4. Ethernet/IP parameters of the laboratory setup modules

	CompactLogix L35E	1734-AENT	PanelView Plus 600	PowerFlex40	WAGO 750- 341
IP Address	192.168.1.1	192.168.1.2	192.168.1.3	192.168.1.5	192.168.1.182
Subnet Mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Gateway IP Address	none	none	none	none	none

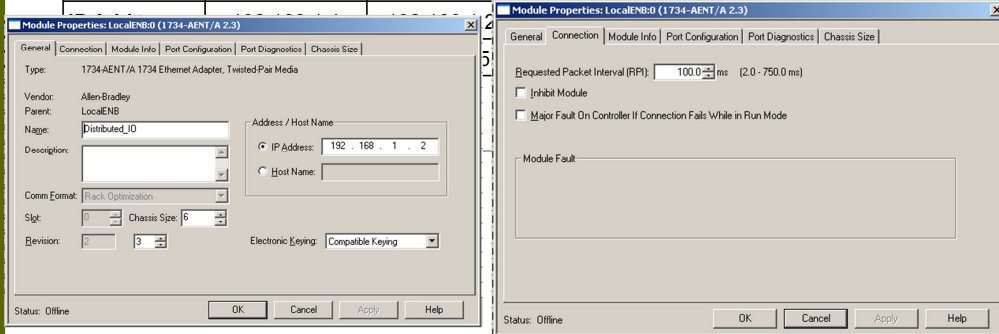
The Ethernet/IP parameters assigned to the each node are collected in Table 5.4.

EtherNet/IP Configuration

Configuration is performed using RSLinx and RSLogix 5000 software.

Table 5.4. Ethernet/IP parameters of the laboratory setup modules

	CompactLogix L35E	1734-AENT	PanelView Plus 600	PowerFlex40	WAGO 750- 341
--	----------------------	-----------	-----------------------	-------------	------------------



© 2011 Adam Pilat, Maciej Rosól, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.1 System architecture>



The Ethernet/IP parameters assigned to the each node are collected in Table 5.4.

▶ Aero-Lift Test Rig

- 4 floor building model.
- Sensors – measure the cart position and detect the cart motion.
- High level control – automata that have 8 inputs and one output.

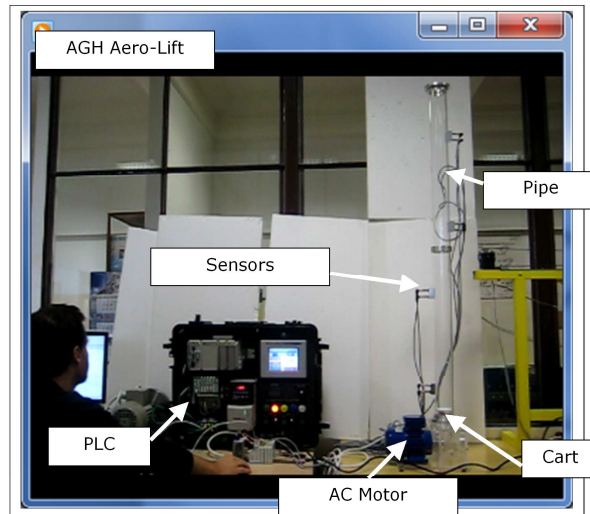


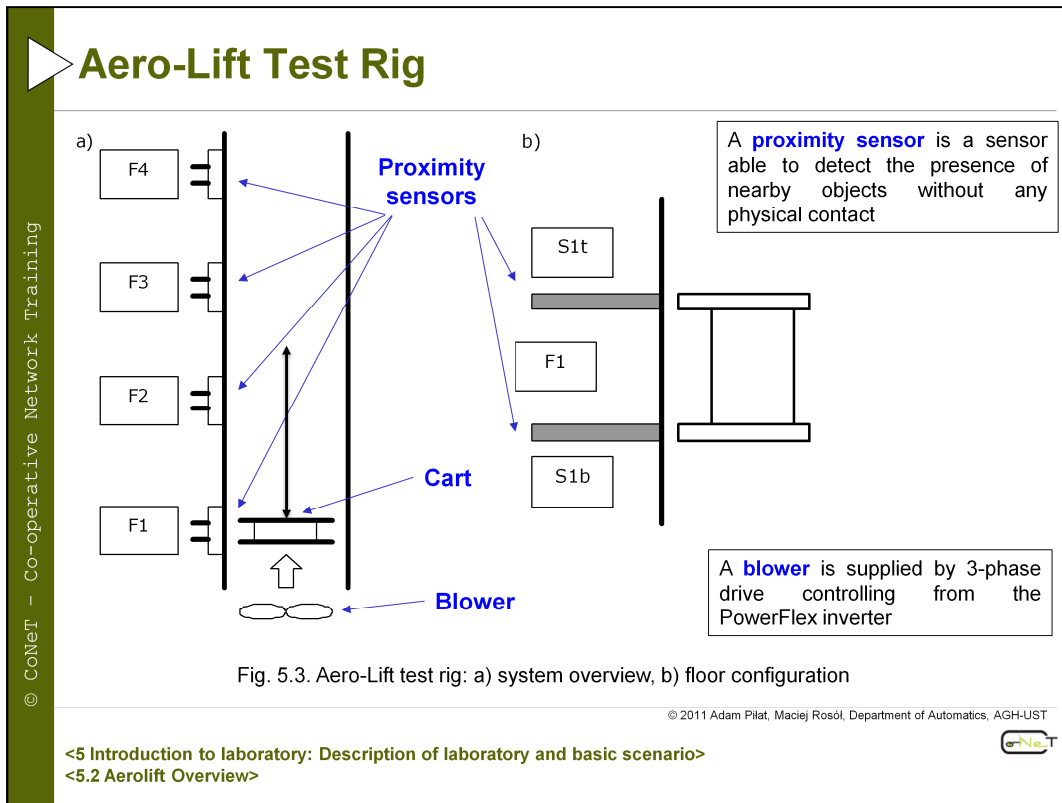
Fig. 5.2. Aero-Lift - laboratory test-rig

© 2011 Adam Pilat, Maciej Rosól, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.2 Aerolift Overview>



The aero-lift (AL) is a system dedicated to a various number of tasks. The test-rig contains the blower based on 3-phase drive steered by the inverter, 2 pipes for vertical motion of the cart and 8 discrete sensor to measure the cart position (see Fig. 5.2).



The system was designed to fulfill requirements of distributed drive control and digital measurements. The pipes are mounted in a stackable form to obtain 4 floor building model. Every floor is equipped with 2 sensors to detect the cart position. The cart contains 2 plates that stabilizes the motion in the pipe and prevents flipping. These plates are used for the sensing purposes too. The schematic diagram of the system and floor section is presented in Fig. 5.3 a and b respectively.

These two sensors are used to position the cart at the desired floor and to detect the cart motion. Observing the history of sensor states the cart motion can be detected. Thus, the controller could provide an appropriate control strategy.

The high level control can be realized as automata that have 8 inputs and one output (as four values) with a steady control adjusted for the desired level.

The direct dynamics controller must be much more complex due to the cart dynamics, nonlinearity of the lifting force that depend on the cart position, its mass, and blower characteristics. Its performance must be investigated in the identification experiments. This controller can operate on the measurement available from sensors or can use an internal observer to estimate the cart position.

Because the aim of the CoNeT project is to diagnose and observe consequences of the lost data, jitter or delays in the data collection, the direct measurement of the cart sensor will be used. Therefore, the digital sensors are connected to 2 devices that can handle digital signals and transmit their states via network to the host PLC. These devices: distributed I/O and WAGO controller are available via the network addresses X.X.X.2 and X.X.X.182 respectively. Both of them can be configured with a variable scan time from 2-750ms from the PLC side.

Basis of the PLC controller

PLC controller is an industrial computer, which works under the real time operation system.

```

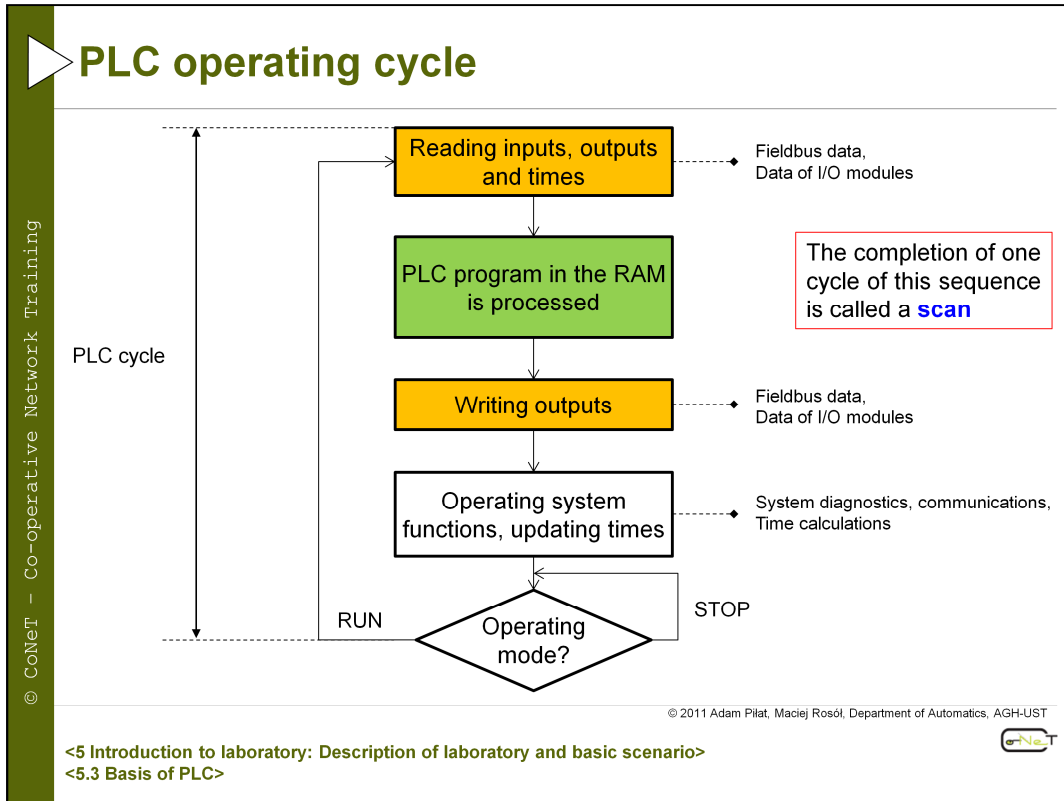
graph TD
    subgraph PLC_Controller [PLC Controller]
        CPU[CPU]
        RAM[RAM/ROM Memories]
        Input[Analog/Digital Input Modules]
        Output[Analog/Digital Output Modules]
        Program[PLC program]
        CPU --- RAM
        Input --> CPU
        CPU --> Output
    end
    Sensors[Sensors] --> Input
    Output --> Actuators[Actuators]
    Program --> CPU
  
```

Fig. 5.4. Components of the PLC system

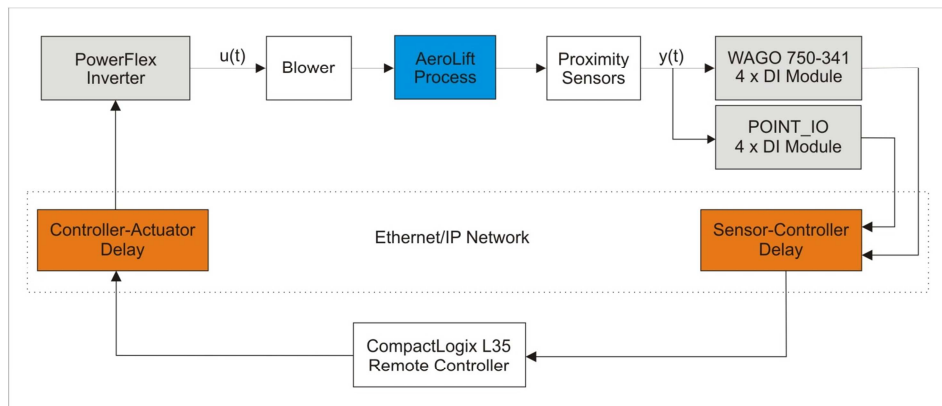
© 2011 Adam Pilat, Maciej Rosól, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.3 Basis of PLC>

The function of an input module is to convert incoming signals into signals, which can be processed by the PLC, and to pass these to the central control unit. The reverse task is performed by an output module. This converts the PLC signal into signals suitable for the actuators. The actual processing of the signals is effected in the central control unit in accordance with the program stored in the memory.



Real-time network structure



The presented structure can be used to demonstrate the performance of the Ethernet/IP protocol and observe propagation time delays between sensor - remote controller and remote controller - actuator.

© 2011 Adam Piłat, Maciej Rosół, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.4 Real-time network structure>

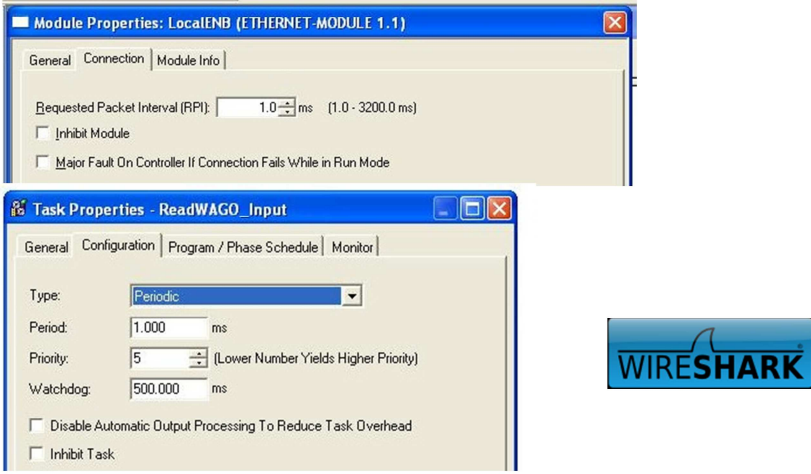


The process is equipped with a sensors and actuator. Measurements and control signals are connected to input/output modules of PLC controller. PLC works as I/O adapter (intelligent adapter) and is connected with Remote controller by Ethernet network.

The presented structure can be used to demonstrate the performance of Ethernet/IP protocol, observe propagation time delays between sensor-remote controller and actuator.

▶ **Configuration User Interface**

Aero-lift – the dynamical system sensitive for sampling time and latency in data transmission



© CoNeT – Co-operative Network Training

Fig. 5.4 User Configuration Interface

© 2011 Adam Pilat, Maciej Rosol, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
< 5.4 Real-time network structure >

Fig. 5.4 presents the user interface that allows to configure the Ethernet module and the Requested Packet Interval (RPI) and schedule for digital data scan with specified task type set to Periodic and defined Period, Priority and Watchdog. The PLC and system programmer has the availability to manually adjust appropriate rating to fulfill the control system requirements. Moreover, it means that the system user can test the influence of the sensing frequency for the system controllability. Moreover, the sensed data are transmitted via the network and some latency in the transmission can occur. This is observable as a jitter in the data delivery and can be diagnosed via the appropriate software. To diagnose these effect the WireShark application can be used.

The aero lift is an example of the dynamical system that is sensitive for sampling time and latency in data transmission. With this set-up a number of consequences caused by the inaccurate settings for Ethernet modules and I/O blocks can be observed. The extra protection algorithms can be implemented to guarantee the safe operation of the lift.



Real-time network structure

What are the parameters deciding about a network control system quality and performance?

1. Scan time of a PLC.
2. Period time of an implemented Periodic Task.
3. Requested Packet Interval (RPI) time.
4. Latency in a data transmission.

Possible problems to be solved on the laboratory setup:

1. Analyzing an influence of the sampling frequency and/or the RPI on the AeroLift stability and controllability.
2. Determination of a propagation time delays between sensor-remote controller and actuator in the Ethernet/IP network based on packet analysis.
3. Analyzing an influence of the latency time, observable as a jitter in the data delivery, on performance and quality of the control network system (to diagnose these effect the WireShark application can be used).

© 2011 Adam Pilat, Maciej Rosól, Department of Automatics, AGH-UST

<5 Introduction to laboratory: Description of laboratory and basic scenario>
<5.4 Real-time network structure>





References

- [1] Allen Bradley, RSLogix5000. Programming Software, Version 16.03
- [2] Allen Bradley, RSLinx Classic. Getting Results Guide, PUBLICATION LINX-GR001G-EN-E, September 2010.
- [3] Allen Bradley, Logix5000 Controllers I/O and Tag Data. Programming Manual, Publication 1756-PM004A-EN-P, July 2007.
- [4] WAGO Kontakttechnik GmbH & Co. KG, 750-341 Modular ETHERNET TCP/IP I/O-System. User's Manual, Ver. 1.1.1, Germany, 2007.
- [5] WAGO Kontakttechnik GmbH & Co. KG, Using the WAGO 750-341 as Remote I/O with a ControlLogix Ethernet/IP Bridge Module, Application Note, Germany, 2004.

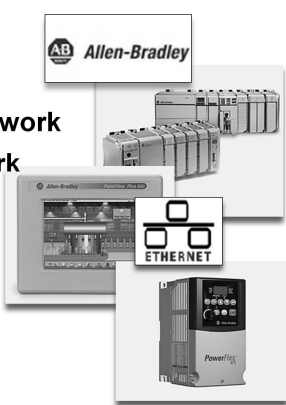


- [1] Allen Bradley, RSLogix5000. Programming Software, Version 16.03
- [2] Allen Bradley, RSLinx Classic. Getting Results Guide, PUBLICATION LINX-GR001G-EN-E, September 2010.
- [3] Allen Bradley, Logix5000 Controllers I/O and Tag Data. Programming Manual, Publication 1756-PM004A-EN-P, July 2007.
- [4] WAGO Kontakttechnik GmbH & Co. KG, 750-341 Modular ETHERNET TCP/IP I/O-System. User's Manual, Ver. 1.1.1, Germany, 2007.
- [5] WAGO Kontakttechnik GmbH & Co. KG, Using the WAGO 750-341 as Remote I/O with a ControlLogix Ethernet/IP Bridge Module, Application Note, Germany, 2004.

Lesson „EtherNet/IP on Allen-Bradley platform”


CoNeT Mobile Lab: EtherNet/IP on Allen-Bradley platform

- 1 Distributed control architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory:
Description of laboratory and basic scenario
- 6 Introduction to laboratory: Software tools



© CoNeT – Co-operative Network Training

Co-operative Network Training



CoNet Mobile Lab: EtherNet/IP on Allen Bradley platform

Introduction

- 1 Distributed Control Architecture
- 2 Real-time control system and real-time network
- 3 Monitoring and testing the Ethernet network
- 4 Introduction to EtherNet/IP technology
- 5 Introduction to laboratory: Description of laboratory and basic scenario
- 6 Software tools

© 2011 Wojciech Grega, Department of Automatics, AGH University of Science and Technology

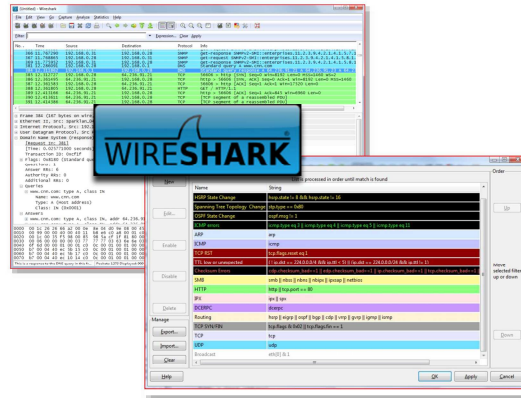
<6 Introduction to laboratory: Software tools>

6.1 Wireshark – Network Sniffer

6.2 Wireshark Features

6.3 Tutorial

6.4 References



© CoNeT – Co-operative Network Training

© 2011 Tomasz Mikis, Department of Automatics, AGH-UST

Content of the lesson „EtherNet/IP on Allen-Bradley platform”



6.1 Wireshark – network sniffer

6.2 Wireshark features

6.3 Tutorial

6.4 References

© CoNeT – Co-operative Network Training

▶ Wireshark - Introduction

Freeware network sniffer – administration tool

command menus

display filter specification

listing of captured packets

details of selected packet header

packet content in hexadecimal and ASCII

The screenshot shows the Wireshark interface. At the top, there's a menu bar and toolbar. Below that is a display filter field. The main area is a list of captured packets with columns for No., Time, Source, Destination, Protocol, and Info. The selected packet (No. 4) is expanded in the details pane, showing Ethernet II, Internet Protocol, and Hypertext Transfer Protocol layers. The bottom pane shows the raw packet data in hexadecimal and ASCII.

Fig. 6.1 Wireshark GUI.

© 2011 Tomasz Miklis, Department of Automatics, AGH-UST

<6 Introduction to laboratory: Software tools>


<6.1 Wireshark – Network Sniffer>

Wireshark (originally called Ethereal) is a freeware network sniffer. Sniffer investigates and analyzes the network traffic. It allows administrators to have the opportunity to recognize weaknesses and vulnerabilities in the network and quickly find a way to resolve the problems.

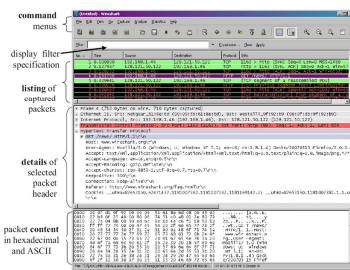
Wireshark is very similar to *tcpdump* (common packet analyzer that runs under the command line), but Wireshark has a graphical front-end, and many more information sorting and filtering options. Wireshark allows the user to see all traffic being passed over the network by putting the network interface into promiscuous mode (configuration of a network card that makes the card pass all traffic it receives to the central processing unit rather than just frames addressed to it).

© CoNeT – Co-operative Network Training


▶ Wireshark - Features

 features:

- Captures Data "from the wire" or from a file
- Reads Data from a number of types of networks,
- GUI or the terminal (command line) interface
- Operations on files
- Display filter.
- Plug-ins can be created for dissecting new protocols.
- VoIP calls detection
- Raw USB traffic (under Linux).



© 2011 Tomasz Mikis, Department of Automatics, AGH-UST

<6 Introduction to laboratory: Software tools>


<6.2 Wireshark Features>

Wireshark is software that "understands" the structure of different networking protocols. Thus, it is able to display the encapsulation and the fields along with their meanings of different packets specified by different networking protocols. Wireshark uses packet capture (pcap) to capture packets, so it can only capture the packets on the types of networks that pcap supports.

- Data can be captured "from the wire" from a live network connection or read from a file that recorded already-captured packets.
- Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.
- Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, tshark.
- Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.
- Data display can be refined using a display filter.
- Plug-ins can be created for dissecting new protocols.
- VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding the media flow can even be played.
- Raw USB traffic can be captured with Wireshark. This feature is currently available only under Linux.

▶ Wireshark - Tutorial

© CoNeT – Co-operative Network Training

1. Download and install the software. The latest version (1.4.1) is available for download from the official site <http://www.wireshark.org/download.html>.
2. Once Wireshark is installed, start it up and you'll be presented with the blank screen shown below.

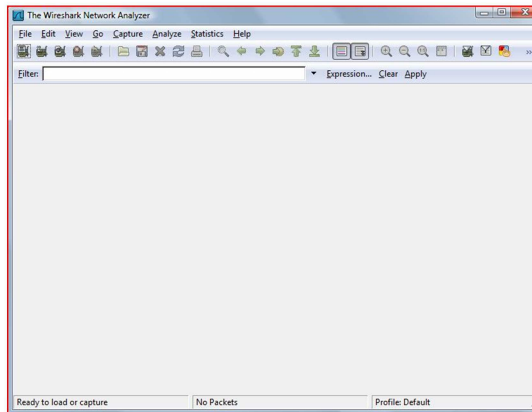


Fig. 6.2 Wireshark GUI – blank screen.

© 2011 Tomasz Miklis, Department of Automatics, AGH-UST

<6 Introduction to laboratory: Software tools>
<6.3 Tutorial>



1. Download and install the software. The latest version (1.4.1) is available for download from the official site <http://www.wireshark.org/download.html>
2. Once Wireshark is installed, start it up and you'll be presented with the blank screen shown in Fig 6.2.

Tutorial – Capture Configuration

3. To start scanning, choose Interfaces from the Capture menu. You'll see a pop-up window similar to the one below

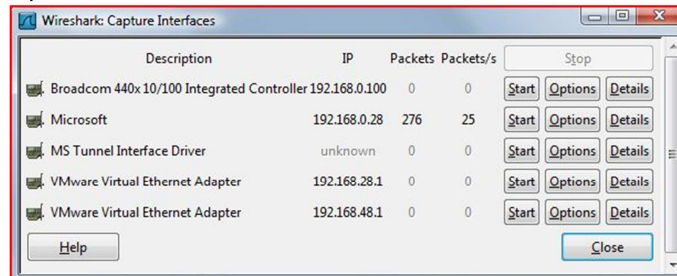


Fig. 6.3 Wireshark Capture Interface

Options button – configuration of the advanced options.

© 2011 Tomasz Miklis, Department of Automatics, AGH-UST

<6 Introduction to laboratory: Software tools>
<6.3 Tutorial>



3. To start scanning, choose Interfaces from the Capture menu. You'll see a pop-up window similar to the one in Fig. 6.3

If you'd like to configure advanced options -- like capturing a file, resolving MAC addresses and DNS names, or limiting the time or size of the capture -- click the Options button corresponding to the interface you wish to configure. Many of these options can help to improve the performance of Wireshark. For example, you can adjust settings to avoid name-resolution issues, as they will otherwise slow down your capture system and generate large numbers of name queries. Time and size limits can also place limitations on unattended captures. Otherwise, simply click the Start button next to the name of the interface on which you wish to capture traffic.

Wireshark Sniffing Interface

4. The Wireshark screen will immediately begin filling up with traffic seen on the network interface, as shown below

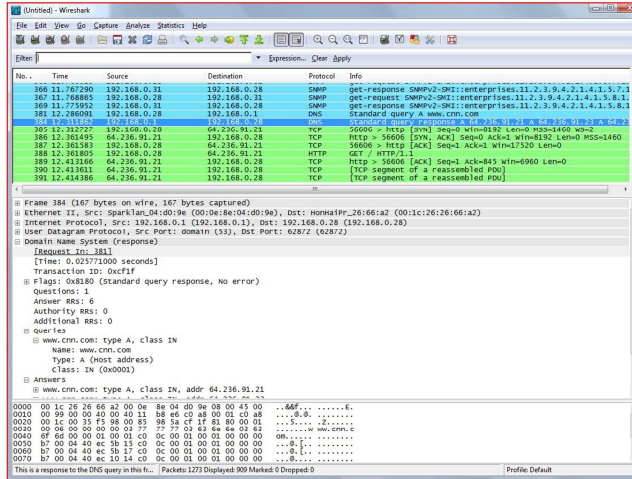


Fig. 6.4 Wireshark Screen

© 2011 Tomasz Miklis, Department of Automatics, AGH-UST

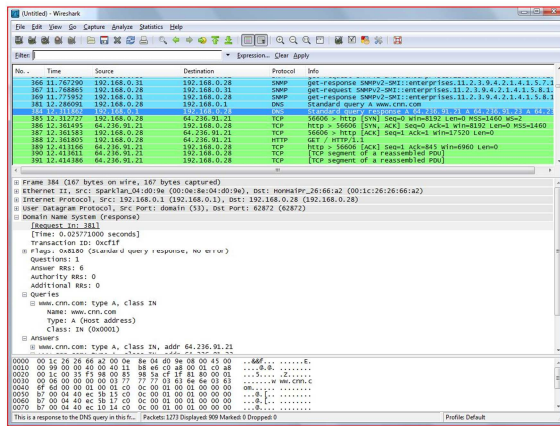
<6 Introduction to laboratory: Software tools>
<6.3 Tutorial>



4. The Wireshark screen will immediately begin filling up with traffic seen on the network interface, as shown in Fig. 6.4

Wireshark Sniffing Interface

© CoNeT – Co-operative Network Training



- Each line in the top pane of the Wireshark window corresponds to a single packet,
- The middle pane contains drill-down details on the packet selected in the top frame.

© 2011 Tomasz Miklis, Department of Automatics, AGH-UST



<6 Introduction to laboratory: Software tools>
<6.3 Tutorial>

Each line in the top pane of the Wireshark window corresponds to a single packet seen on the network. The default display shows the time of the packet (relative to the initiation of the capture), the source and destination IP addresses, the protocol used and some information about the packet. You can drill down and obtain more information by clicking on a row. This causes the bottom two window panes to fill with information.

The middle pane contains drill-down details on the packet selected in the top frame. The "+" icons reveal varying levels of detail about each layer of information contained within the packet. In the example above, In this example a DNS response packet was selected. The DNS response (application layer) section of the packet is expanded to show that the original was requesting a DNS resolution for www.cnn.com, and this response is informing us that the available IP addresses include 64.236.91.21. The bottom window pane shows the contents of the packet in both hexadecimal and ASCII representations.

Color Schemes

5. Color is your friend when analyzing packets with Wireshark.

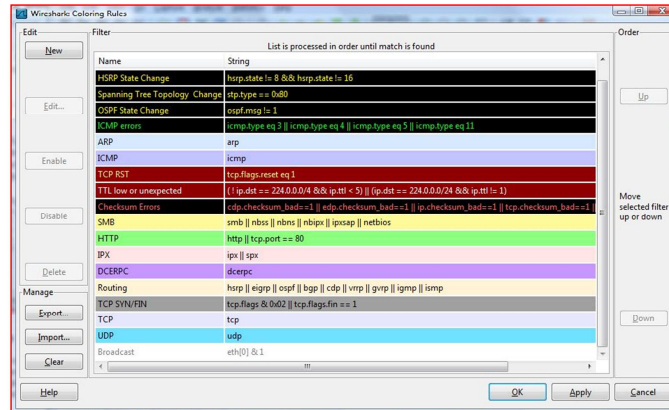


Fig. 6.5 Wireshark color-coding scheme window.

© 2011 Tomasz Miklis, Department of Automatics, AGH-UST

<6 Introduction to laboratory: Software tools>
<6.3 Tutorial>



5. Color is your friend when analyzing packets with Wireshark. Notice in the example above that each row is color-coded. The darker blue rows correspond to DNS traffic, the lighter blue rows are UDP SNMP traffic, and the green rows signify HTTP traffic. Wireshark includes a complex color-coding scheme (which you can customize). The default settings appear below.

References

- [1] Orebaugh, Angela; Ramirez, Gilbert; Beale, Jay (February 14, 2007). Wireshark & Ethernet Network Protocol Analyzer Toolkit. Syngress. p. 448. ISBN 1597490733. <http://www.syngress.com/hacking-and-penetration-testing/Wireshark-amp-Ethereal-Network-Protocol-Analyzer-Toolkit/>
- [2] Sanders, Chris (May 23, 2007). Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems. No Starch Press. p. 192. ISBN 1593271492. <http://nostarch.com/packet.htm>
- [3] Chappell, Laura (March 31, 2010). Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide. Protocol Analysis Institute, dba "Chappell University". p. 800. ISBN 1893939995



References:

- [1] Orebaugh, Angela; Ramirez, Gilbert; Beale, Jay (February 14, 2007). Wireshark & Ethernet Network Protocol Analyzer Toolkit. Syngress. p. 448. ISBN 1597490733. <http://www.syngress.com/hacking-and-penetration-testing/Wireshark-amp-Ethereal-Network-Protocol-Analyzer-Toolkit/>
- [2] Sanders, Chris (May 23, 2007). Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems. No Starch Press. p. 192. ISBN 1593271492. <http://nostarch.com/packet.htm>
- [3] Chappell, Laura (March 31, 2010). Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide. Protocol Analysis Institute, dba "Chappell University". p. 800. ISBN 1893939995