International Journal of Innovative Computing, Information and Control
Volume **10**, Number **4**, August **2014**

ICIC International ©2014 ISSN 1349-4198
pp. 1275–1289

ADVANCES IN DISTRIBUTED CONTROL FOR FACTORY AUTOMATION ON ETHERNET TECHNOLOGY

Alexandre Baratella Lugli[1], Lucia R. Horta R. Franco[2]
and Max Mauro D. Santos[3]

[1]Department Coordinator of Industrial Automation
National Institute of Technology (INATEL)
Santa Rita Sapucaí, Brazil
baratella@inatel.br

[2]UniRede and Distance Leaning
Federal University of ABC
Santo André, Brazil
lucia.franco@ufabc.edu.br

[3]Department of Industrial Automation
Federal Technological University of Paraná – Ponta Grossa
Ponta Grossa, Brazil
maxsantos@utfpr.edu.br

Received July 2013; revised January 2014

Abstract. *The TCP/IP suite protocol is one of the most widespread protocols groups for communication at long, average and short distances involving computer systems. Then, for trying the industrial networks standardization, for some years, it was also implemented in the industrial environment. The manuscript aims to propose a new way to accomplish the fieldbus modules control, using a flexible distributed architecture applied to the industrial Ethernet networks. The work proposal is to develop an algorithm for messages scheduling, applied to a distributed architecture, with the removal of the master controller, where only the fieldbus modules and switches operate on the network, and then having a communication messages distributed control. The proposed algorithm uses an off-line or pre-run-time type technique of messages scheduling. Therefore, the manuscript develops a new communication concept, applied industrial Ethernet networks, with communication network control and each field element distributed throughout the manufacturing process. For this, it is necessary to model the new communication concept, perform the messages scheduling (which is essential for defining the message communication order in the network trafficked, due to not using a centralized master controller) and perform verification testing and idea proposal validation.*

Keywords: Control system design, Distributed architecture, Industrial Ethernet, Scheduling

1. Introduction.

The protocols network suite, known as Internet standard, is one of the most widespread communication networks architectures to interconnect computer systems. It has been developed by V. G. Cerf and R. E. Khan [1] in the mid seventies, and has been growing since then.

The Ethernet suite protocol is the one that is now part of the Internet suite and is used in the data link and physical layers. R. M. Metcalfe, created it also in the mid seventies, to assist the physical connection of computer systems [2].

The TCP/IP suite protocol is one of the most widespread protocols groups for communication at long, average and short distances involving computer systems. Then, for trying the industrial networks standardization, for some years, it was also implemented in the industrial environment [3,17]. Nowadays, there are fourteen industrial Ethernet networks, as follows: PROFINET, Ethernet/IP, HSE (High Speed Ethernet), Modbus/TCP, EPA, EPL, EtherCAT, IEC 61850, JetSync, P-Net, Sercos III, SynqNet, TCnet and Vnet/IP [3,4,18]. These networks operate typically with master-slave type communication [3-5].

This manuscript discusses one of the most popular communication technologies, which is also applied for distributed systems in industrial environments, known as "industrial Ethernet". The proposal presented seeks to develop a communication methodology adequate for flexible distributed control applications. The manuscript aims to propose a new way to accomplish the fieldbus modules control, using a flexible distributed architecture applied to the industrial Ethernet networks. The manuscript proposal is to develop an algorithm for messages scheduling, applied to a distributed architecture with the removal of the master controller, where only the fieldbus modules and switches operate on the network, and then having a distributed control of the manufacturing process. The proposed algorithm uses an off-line or pre-run-time type technique of messages scheduling.

1.1. **The requirement for a distributed control concept.**

Most difficulties faced by distributed control system designers are related to the network elements definition and their settings, as well as checking the imposed requirements by the physical network (power cables sizing and network devices).

Currently, the main difficulty for the use of industrial networks is related to the interoperability among equipment from different manufacturers. There is a great variety of equipment, cables, connectors and other elements applied to the same industrial network [14].

More than often, designers become dependent on a certain technology due to the lack of standardization in industrial Ethernet networks.

During a project implementation, several problems may occur, but some of them are not viewed during the design phase due to the lack of adequate network simulation resources.

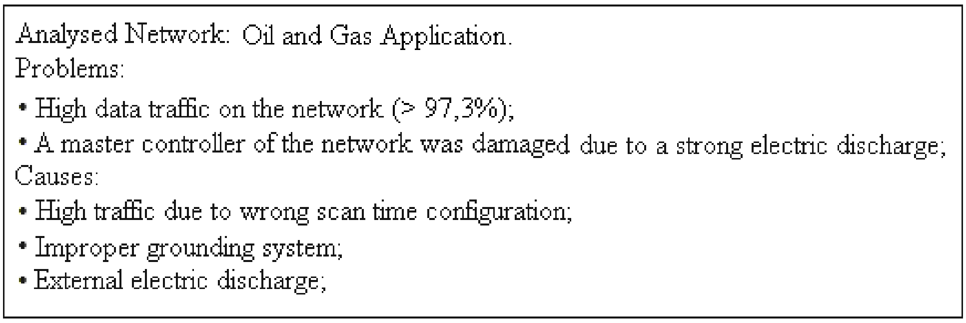
Below are some details which illustrate the problem:

- A long time for starting the plant due to some type of physical network error (inadequate cable lengths or invalid electrical characteristics).
- The network logical project is incorrect (for instance, the scan time is not suitable for the production process), affecting all the process dynamics.
- The side effects of using the selected technology are in general not fully discussed during the network design phase and installation.
- For centralized systems, the distributed system design usually does not consider master controller physical redundancy or the field modules redundancy, due to the difficulty of performing such type of analysis and the high cost to achieve this redundancy.

Another problem encountered by industrial network designers is the system control logic centralization at a single point: the network master controller. Most fieldbuses have this centralized architecture. This may cause problems to designers in the case of master controller failures, for instance, the cable rupture or power supply defaults.

Figure 1 illustrates, in practice, some of the problems mentioned above regarding a set-up industrial network. The studied cases are highlighted through the use of technical reports prepared by a company in the industrial networks field, during a technical visit to a client [6].

The outcome of these technical reports, would be helpful to the network designers to have a graphical environment allowing them to verify the configured parameters and to make the required network simulations, and verifications, before starting the installation



Analysed Network: Oil and Gas Application.
Problems:
- High data traffic on the network (> 97,3%);
- A master controller of the network was damaged due to a strong electric discharge;

Causes:
- High traffic due to wrong scan time configuration;
- Improper grounding system;
- External electric discharge;

Figure 1. Manufacturer factory example

and the network devices configuration. Furthermore, it is essential to guarantee to schedule that the control messages are adequately distributed along the field modules and not concentrated on a single network location (master controller).

Some works carried out, [9,11,21], have proposed, in some way, the solution to these problems. However, the solutions are not completely resolved from the designer viewpoint and the applicability to industrial Ethernet networks [20,21]. The papers [20,21] discuss a modeling and control system for industrial Ethernet networks, with centralized control in the master controller network, not showing or mentioning the proposal for a distributed control applied to industrial Ethernet networks. Moreover, the works [11,19] have already proposed, in some way, using the techniques of scheduling messages applied to Foundation Fieldbus network to perform distributed control on the bus, but did not address the treatment of faults that may occur in field modules.

This manuscript aims to develop a new approach for industrial Ethernet networks design, considering both: the non centralized control communication and a set of field elements distributed throughout the manufacturing process. The main steps for this novel approach implementation are the need to model the communication architecture, to perform the scheduling (which is essential to define the order of the messages being transferred on the bus, due to the non centralized master controller) and to perform the verification, testing and validation of the proposed idea.

2. **The Proposed Approach.** The fulfillment to time constraints of the scheduling system, applied to industrial Ethernet networks ensures that, when a message is trasnmitted, it must be completed before its respective expiration time (deadline time). This paradigm must be respected by the transferred messages that are related to the distributed control applications, since a non-compliance may lead to a communication failure, causing the messages loss. It is of fundamental importance to establish a precedence relation or execution order to the messages that must be transmitted on the network. In addition, the proposed solution to distributed control, using the RM (Rate Monotonic) scheduling technique, operates with non-preemptive messages, where the messages cannot be interrupted by another message being transferred. The cyclic messages (e.g., sensors reading and actuators states updating) and acyclic (e.g., a field module parameter or possible network fault) should be initiated and completed in the same execution time for deterministic guarantee and for a feasible performance on the industrial Ethernet network communication [9-11].

With the finalization of the scheduling algorithm execution, it obtains a schedule of actions to be accomplished, specified through a temporary forecast defined according to the messages characteristics and priorities, so that, during the communication activity on the bus network, these processes assist the communication time consistence.

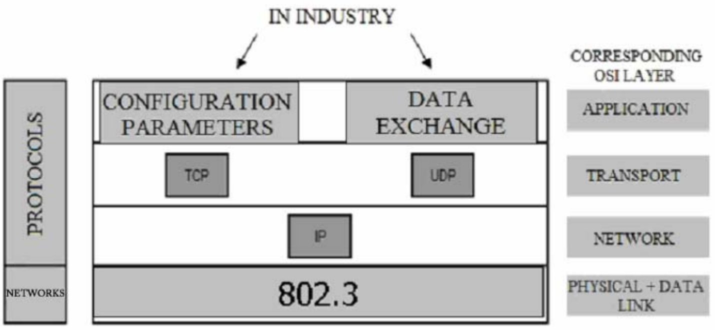


Figure 2. Layers of protocols from an industrial networks architecture [7,8]

The proposed solution in this paper uses an off-line scheduling (pre-run-time technique, type RM), with previous priority in each transferred message on the bus, which is non-preemptive. With the scheduling table generated after the running scheduler a time schedule of actions is obtained to be performed on the bus through a specified prediction time defined according to the transferred messages characteristics and priorities. Therefore, it can be inserted into one or more network field modules to perform the bus communication control, to guarantee the distributed control realization applied to industrial Ethernet networks. Then, the inclusion of this scheduling table in more than one field module ensures the proposed architecture operation, even in case of failure at any field module.

Figure 2 illustrates the concept of layers division applied to a typical industrial Ethernet network [7,8]. In Figure 2 it is possible to notice that the cyclical messages timing (information exchanges between sensors and actuators on the network) uses UDP, due to the imposed timing requirements by the distributed control application. The TCP protocol is used for acyclic messages, because the timing requirements of the transferred information are not critical in this case.

The network design approach proposed in this paper uses RM technique [9,10], where the scheduling task is divided into three different implementation levels, according to the prioritization of the messages transferred on the bus. Therefore, the developed scheduler analyzes the set of messages that need to be transmitted from each field module in three levels, to sort the information flow on the network. At the first level, the scheduler groups all messages from each field module in "*sub-groups by priority*", which are defined according to the message priorities (three different priority levels have been proposed: from zero up to two).

At the second level, the scheduler analyzes the total execution time of each module, in each field "*sub-priority group*", creating another sub-group within the first level, called "*sub-groups by runtime*" and, then arranges all the messages of each field module, according to the calculated time. In the third and final level, the scheduler combines all the messages in each "*sub-group by the runtime*", in ascending order.

Figure 3 illustrates the priority scheduler concept developed for the distributed control implementation approach proposed for industrial Ethernet networks.

In Figure 3, there are ten messages to be transmitted over the communication network among seven different field modules, where a letter that indicates the message is being transferred, coming from a field module to the network, as addressed below, represents each message:

1. Square I from field module one.
2. Square B from field module two.

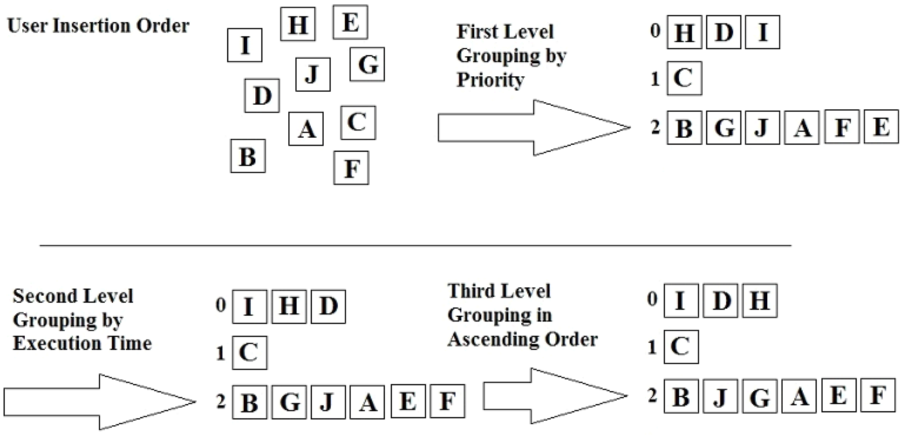


Figure 3. The scheduler strategy considering three level grouping

3. Squares H and D from field module three.
4. Squares G, A and J from field module four.
5. Square F from field module five.
6. Square E from field module six.
7. Square C from field module seven.

The traffic priorities of each message have been pre-defined through a configuration setup run at each field module.

For the example of Figure 3, the timing parameters for the field modules are sent according to the logic described below:

1. The runtime field module three is less than the field module one.
2. The runtime field module four is less than the field module two.
3. The runtime field module five is less than the field module six.
4. The runtime field module which has message H is less than the field module which has message D.
5. The runtime field module which has message A is less than the field module which has message G, and this is less than the field module which has message J.

Then, as final result for the proposed approach to deal with message scheduling in Ethernet networks (RM type scheduling), there is the condition illustrated in Figure 4, where message H is the first to be transferred by the network and message E is the last one.

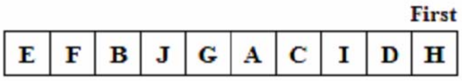
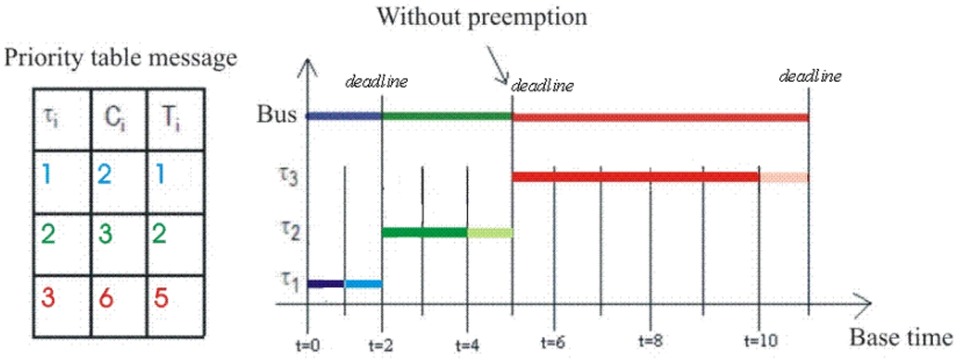


Figure 4. Execution order of the messages after the scheduling

Regarding the priorities of messages to be scheduled on the bus (the first level scheduler), there is a message assignment in a fixed, non-preemptive order, as follows:

- Transferred messages within the same field module (a field module sends the information to itself). For this case, the algorithm classifies the message priority as zero.
- Transferred messages within different field modules (a field module sends the information to another one, i.e., a beginning/ending type of message). For this case, the algorithm classifies the message priority as one.



Priority table message

τ_i	C_i	T_i
1	2	1
2	3	2
3	6	5

FIGURE 5. The rate monotonic as scheduling strategy purposed

- Intermediate messages, i.e., messages that depend on previous information before sending it to another field module (a field module receives the message and forwards it to the other one). For this case, the algorithm classifies the message priority as two.

In addition, the messages transferred at the mentioned scheduling information proposal include cyclic and acyclic messages. Then, each message transferred at the proposed network will have a cyclic and acyclic time communication percentage.

Figure 5 illustrates a proposed RM scheduling, for a network containing three messages to be transferred, according to the priorities shown at the table (one, two and three, respectively).

Consider a set of periodic tasks with the following properties and definitions in Equation (1).

$$\tau_i = \tau_i(T_i; C_i; R_i) \tag{1}$$

where:

- τ_i: Message "i" to be staggered.
- C_i: Maximum time for message "i" implementation.
- T_i: Actual runtime message "i".
- R_i: Message "i" release time or trigger.

According to that defined in [9], it is possible to use the notation from Equation (1) to represent the RM scheduling; the representation of Figure 5 is: $\tau_1 = \tau_1(1; 2; 0), \tau_2 = \tau_2(2; 3; 2)$ and $\tau_3 = \tau_3(5; 6; 5)$.

For the distributed control application, besides using the scheduling techniques to schedule the messages within the industrial Ethernet network, it is necessary to equate the temporary system to the accomplishment of the messages traffic, allowing the distributed control execution. Equations (2) to (7) illustrate the timing equations of the distributed architecture.

The T_{out} in Equation (2) represents the message time in communication bus and T_i in Equation (3) define the message execution time for each field module (RTU – Remote Terminal Unit).

$$T_{out} = \frac{[(U_{out} + 54.8)]}{(T_{tx})} \tag{2}$$

$$T_i = (T_{out} + T_p + T_{dead}) \tag{3}$$

Equations (4) and (5) present respectively the bus load time and the idle time. Equations (6) and (7) being for acyclic and cyclic messages over bus communication define the bus scheduling accord message properties.

$$\tau = \sum_{i=1}^{n} T_i \tag{4}$$

$$\beta = (T_{scan} - \tau) \tag{5}$$

$$T_{ai} = (C_i - T_i) \tag{6}$$

$$T_{cy} = T_i \tag{7}$$

where:

1. T_{out}: duration to perform one message sending of “U_{out}” bytes size on the network.
2. T_p and T_{dead}: internal field module settings, indicating, respectively, the time spent by the module to handle all its messages and time delay regarding the industrial switch communication.
3. T_i: duration to perform one message sending of “U_{out}” bytes size on the network (execution time), considering the T_p and T_{dead} time delays.
4. C_i: maximum duration (deadline) to perform the sending of only one message of “U_{out}” bytes size on the network, considering the T_p and T_{dead} time delays.
5. R_i: release time of each message to be transferred.
6. i: messages number to be executed.
7. τ: all execution times sum of all field modules.
8. β: total remaining time in communication for future expansion on the bus.
9. T_{tx} and T_{scan}: network global settings, indicating the transmission rate and the scan time, respectively.
10. T_{cy} and T_{ai}: time to cyclic and acyclic communications, respectively, to each message on the network.

According to Lee et al. [13], Popp and Webber [14] and Held [15], there can be performance approximation of the switch delay time to 9.6μs, between each element that is exchanging information on the network and switch (switch communication delay, or buffer). It is an average value, empirically calculated by the authors, which is called communication dead time. In Appendix A there is the illustration of the algorithm flowchart to the proposed solution.

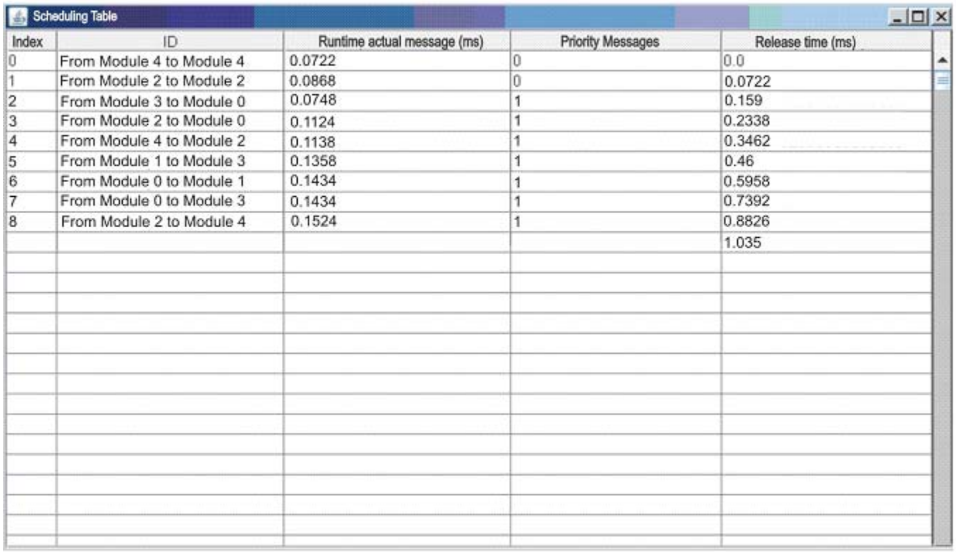
3. **A Novel Tool for Timing Analysis.** In order to perform a timing analysis of the proposed distributed control system, a computational tool has been designed to validate the proposed concept.

Scenario 1: The aim of this scenario is to consider the distributed control logic and to verify some elements of the proposed network below.

To perform the scenario validation, the under mentioned proposed conditions must be viewed:

1. Elements in the simulated network: two switches, each one with four ports, and five field modules.
2. Global parameters configured by the application designer based on their time scenario requirements: transmission rate of 10Mbps, total scan time application of 10ms, five field modules and two four-port switches.

Figure 6 illustrates the network scheduling table for the proposed scenario, with five field modules, nine scheduling messages on the bus, and the distributed control among them. The table shows the staggered communication order to the proposed scenario, according to the settings performed in each field module and the time constraints imposed by the scheduler algorithm where two of these field modules have a dual function, besides



Index	ID	Runtime actual message (ms)	Priority Messages	Release time (ms)
0	From Module 4 to Module 4	0.0722	0	0.0
1	From Module 2 to Module 2	0.0868	0	0.0722
2	From Module 3 to Module 0	0.0748	1	0.159
3	From Module 2 to Module 0	0.1124	1	0.2338
4	From Module 4 to Module 2	0.1138	1	0.3462
5	From Module 1 to Module 3	0.1358	1	0.46
6	From Module 0 to Module 1	0.1434	1	0.5958
7	From Module 0 to Module 3	0.1434	1	0.7392
8	From Module 2 to Module 4	0.1524	1	0.8826
				1.035

FIGURE 6. Scheduling table to Scenario 1

executing the control strategy. They perform simultaneously: the scheduler control of the proposed solution after previous sequence definition to be executed. The field module two performs the control, being the field module zero the reserve in the case of a failure of the main module.

The solution shows the scheduling process in each field module using the *Branch-and-Bound* [19] technique. The *Branch-and-Bound* technique researches the best solution to a search problem [19]. Each time message is defined as the release time minus the deadline time. The difference times between messages are below.

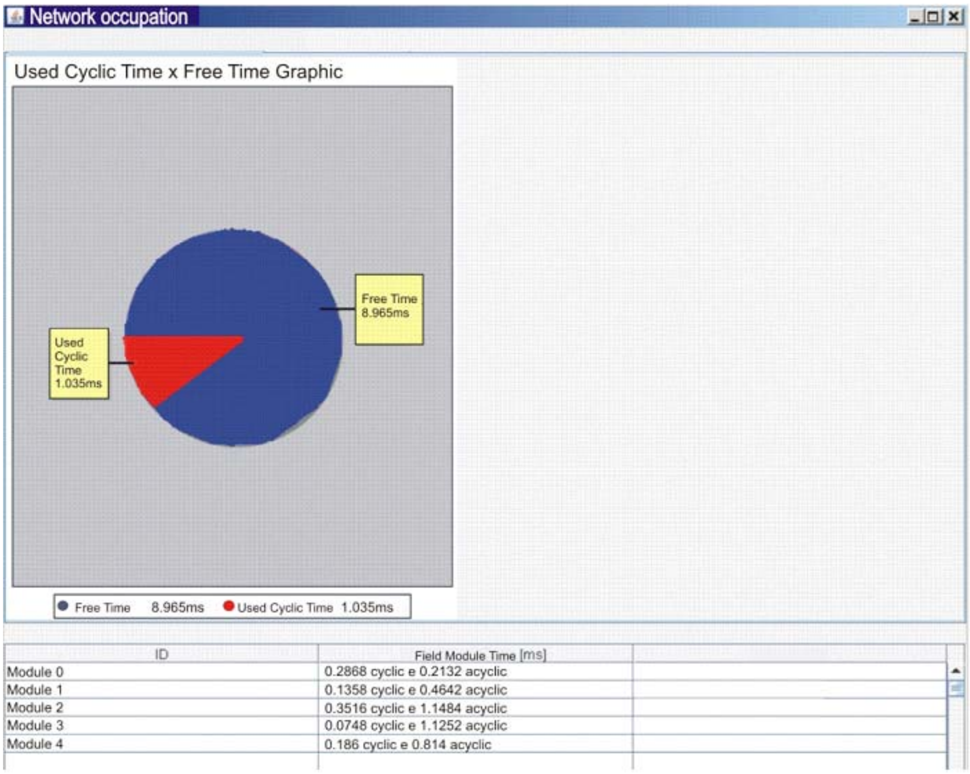
1. Index 0: –0,928ms.
2. Index 1: –1,4132ms.
3. Index 2: –1,1252ms.
4. Index 3: –1,3876ms.
5. Index 4: –0,8862ms.
6. Index 5: –0,4642ms.
7. Index 6: –0,3566ms.
8. Index 7: –0,3566ms.
9. Index 8: –1,3476ms.

It is possible to observe that the solution to messages scheduling in each field module is shown below.

1. Module 0: [Index 6: starts in 0ms; Index 7: starts in 0,072ms].
2. Module 1: [Index 5: starts in 0,159ms].
3. Module 2: [Index 1: starts in 0,9226ms; Index 8: starts in 0,6954ms; Index 3: starts in 0,7702ms].
4. Module 3: [Index 2: starts in 0,5816ms].
5. Module 4: [Index 0: starts in 0,4458ms; Index 4: starts in 0,3024ms].

Figure 7 analyzes the total timing consumed in this scenario, showing the total and individual timing sums, cyclic and acyclic times, from each field module.

Scenario 2: The aim of this scenario is to validate the whole distributed control concept, and also the designed computational tool. The validation is done through a real application that serves for a comparative basis. Then, it is possible to prove that the architecture with distributed control can be applied to an industrial Ethernet network.



ID	Field Module Time [ms]	
Module 0	0.2868 cyclic e 0.2132 acyclic	
Module 1	0.1358 cyclic e 0.4642 acyclic	
Module 2	0.3516 cyclic e 1.1484 acyclic	
Module 3	0.0748 cyclic e 1.1252 acyclic	
Module 4	0.186 cyclic e 0.814 acyclic	

FIGURE 7. Distribution of **Cycle Time x Idle Time** in Scenario 1

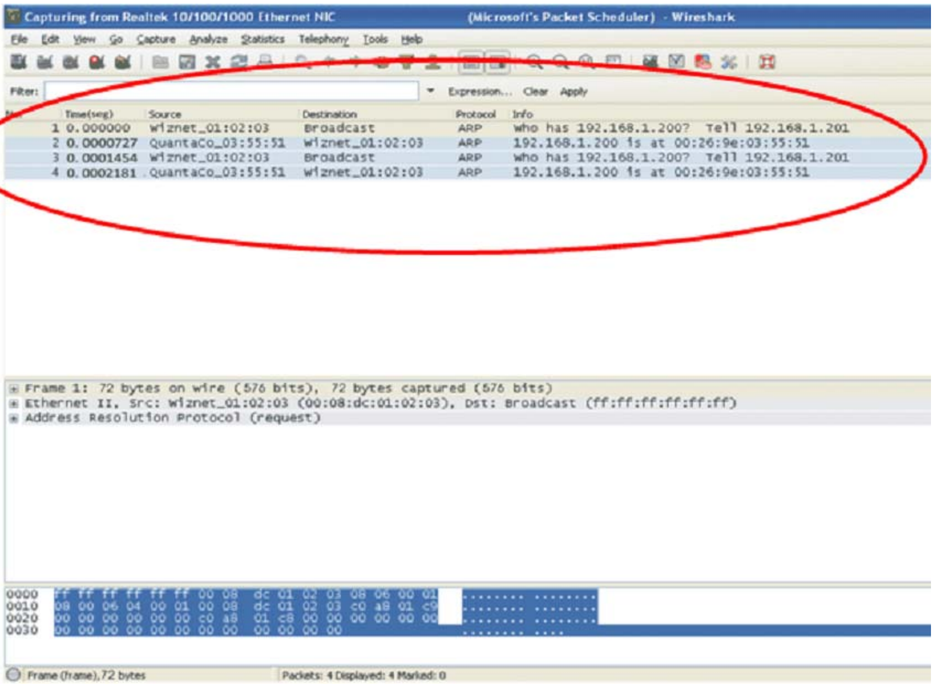
The experimental development was carried out with two field modules, the structure for an industrial Ethernet network, with distributed control, a switch configured with scan time of 1ms and a transmission rate of 10Mbps. The reading and writing bytes represent the temperature sensors and real actuators connected to the module and they contain six input bytes, and six output bytes for each field module.

The packet communication between the two real field modules, with distributed control, can be seen in Figure 8, using a specific computational tool for capturing the messages in the Ethernet network. Then, this capture tool shows the communication timing between each message, each field module, being also possible to display the total communication system timing [16]. This timing, which is similar to the cyclic communication timing from all messages transferred on the network, is $\mathbf{145.4\mu s}$.

Figure 9 illustrates the same information listed above, but it is shown through the computational tool and the proposed architecture (developed to simulate the distributed control). It is possible to view the messages scheduling table between the two field modules and total cyclic timing graph for the field modules control strategies. The total cyclic communication timing used for this scenario is $\mathbf{144.4\mu s}$.

The second scenario illustrates the cyclic scan time calculated by the developed computational tool, being $\mathbf{144.4\mu s}$, and the cyclic scan time measured with the real application and two real field modules, being $\mathbf{145.4\mu s}$. The real cyclic timing measurement was accomplished with the use of a network analyzer in the same network conditions proposed by the computational tool. Therefore, it is possible to conclude that the solution is in accordance with the developed modules, performing the validation of the distributed architecture.

4. **The Genetic Algorithm and Network Optimization.** A genetic algorithm consists of a heuristic search that is based on the biological processes of evolution and looking



```
         Time(seg)  Source              Destination        Protocol  Info
       1 0.000000   Wiznet_01:02:03     Broadcast          ARP       Who has 192.168.1.200?  Tell 192.168.1.201
       2 0.0000727  QuantaCo_03:55:51   Wiznet_01:02:03    ARP       192.168.1.200 is at 00:26:9e:03:55:51
       3 0.0001454  Wiznet_01:02:03     Broadcast          ARP       Who has 192.168.1.200?  Tell 192.168.1.201
       4 0.0002181  QuantaCo_03:55:51   Wiznet_01:02:03    ARP       192.168.1.200 is at 00:26:9e:03:55:51

⊞ Frame 1: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
⊞ Ethernet II, Src: Wiznet_01:02:03 (00:08:dc:01:02:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 08  dc 01 02 03 08 06 00 01   ........ ........
0010  08 00 06 04 00 01 00 08  dc 01 02 03 c0 a8 01 c9   ........ ........
0020  00 00 00 00 00 00 c0 a8  01 c8 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....

Frame (frame), 72 bytes     Packets: 4 Displayed: 4 Marked: 0
```

FIGURE 8. Time analysis from a module in Scenario 2

Index	ID	Runtime actual message (ms)	Priority Messages	Release time (ms)
0	From Module 0 to Module 1	0.0722	1	0.0
1	From Module 1 to Module 0	0.0722	1	0.0722
				0.1444

FIGURE 9. Computational tool simulation to Scenario 2

for solutions to problems of search and optimization. It uses evolutionary processes, such as cross-chromosomes, reproduction, selection of the best partner mutations and so forth, to generate solutions. This type of algorithm is part of a larger class of algorithms, called evolutionary [7].

Genetic algorithms work with the concept of population, represented by a vector of n positions. Each position in the array represents an individual, which corresponds to a representation and a possible solution to the problem. A vector that contains the information that defines chromosomes also represents each individual. Chromosomes are a coded representation of the problem and its efficiency as a possible solution is assessed by an evaluation function. This function provides a value for each chromosome that represents how close he is to the optimal solution.

Other processes used by the genetic algorithms are crossing and mutation of chromosomes.

The task of the process of crossing chromosomes is performing combination of two or more chromosomes, with the aim of generating new individuals that represent a better solution to the problem. However, this is not a general rule, considering that the new individuals may also represent worse solutions.

The mutation operator can enter a search space unexplored for the population. It combines an extremely low probability (0.5%) and a number between zero and a value randomly selected at random. If the number is less than the probability associated with the mutation operator will act on the chromosome.

The goal of using genetic algorithm in this work is the optimization of communication in industrial switch ports. Through successive iterations, it is possible that the algorithm identifies better distribution of modules in the network, doing so with all traffic information is optimized. The industrial switches operate with the priorities in each communication port, which ensures determinism in industrial network.

4.1. **Representation of chromosomes.** In this paper, the representation of chromosomes has been created to be directly linked to the number of switches present on the network topology to be explored [7-9].

Each chromosome is represented by an array of positions that is equal to the total number of switches multiplied by four. Thus, the size of the chromosome is defined as the sum of the gates of all switches. Each position in the vector is called a gene, an indivisible part of the chromosome, which is a port of a switch and has a stored value, indicating a physical connection exists with another port of another switch or other network device [7-9].

The representation of the network is determined that the four defined positions in the vector are the connections of each switch port present on the network, as illustrated in Figure 10.

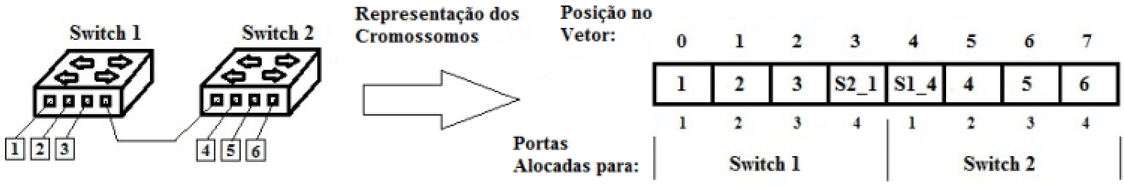


FIGURE 10. Network topology in computational representation of chromosomes

4.2. **Mutation operator.** The mutation operator function is to introduce an unexplored search space to the population.

For this work, a number is drawn to decide whether the mutation operator will act on the chromosome or not. If it is determined that the operator will act, two numbers, which represent two genes on chromosome, are drawn and these positions have changed. When a gene that contains the information for a connection to another switch port, is chosen to be changed, the operation is canceled and another gene is drawn [7].

4.3. **Evaluation function.** The evaluation function can be considered the most important part of a genetic algorithm, as it is built around the knowledge of the problem to be solved [7].

Each chromosome, as previously stated, is a representation of a network, and all should be evaluated. The evaluation is performed through an evaluation function that assigns a score to the solution presented. The closer the score of a chromosome approach the optimal value, the better the solution presented.

For the evaluation of an industrial Ethernet network based on the concept of distributed architecture, you must use the following concepts:

1. Switch Origin: switch that is allocated the device that will send the information.
2. End Switch: switch that is allocated the device that will receive the information.
3. Traffic switch: switch by which information travels to reach the end switch. This switch cannot retain or change the information trafficked.
4. Switches Partners: switches that allocate devices that communicate with each other, but due to physical limitations (number of switch ports) are allocated to different switches.

These concepts are used to set values for the variables that make up the evaluation function. To evaluate the solutions generated are two functions and each has a function for a particular type of situation that may occur on the bus.

Equations (8) and (9) define the evaluation function. Equation (8) evaluates the solutions that have zero local traffic information when all communication between the devices the switch is done outside.

$$\lambda_i = s.n \tag{8}$$

where:

s: variable whose value depends on the ratio between the amount of traffic information among key stakeholders and the total amount of non-trafficked informed on the bus.

n: variable that represents 40 percent of total traffic byte through the bus.

Equation (9) evaluates the solutions that have local traffic information greater than zero.

$$\lambda_i = (\alpha.n1) + (n2s) \tag{9}$$

where:

α: number of bytes that travel locally in the switches of the network.

$n1$: variable that represents 60 percent of total bytes that travel on the bus.

s: variable whose value depends on the relation between the amount of information traffic between switches partners and the total amount of information in a non-trafficked site.

$n2$: variable that represents 40 percent of total traffic bytes across the bus.

4.4. **Result analysis.** There are properties for the evaluation function that analyze the solutions with zero values, if all bus traffic is carried out in a non-local way, and values equal to one hundred, if all traffic is done locally.

Figure 11 illustrates the representation, by a software tool developed, how the bus is represented by the use of chromosomes.

To the bus topology shown in Figure 11 was considered a communication between modules in blue group represented by the switches in the green group. The communication between the modules occurs as described in Table 1.

When analyzing the bus, the genetic algorithm finds all possible routes to this communication become feasible. It is taken as an example for the communication routes between modules 4:07 shown in Table 1.

1. The first route is made feasible module 4 to the switch 4 to switch 0, then the module 7.
2. The second route is performed via the switch 4 to the second switch after the switch to 0, and finally to the module 7.

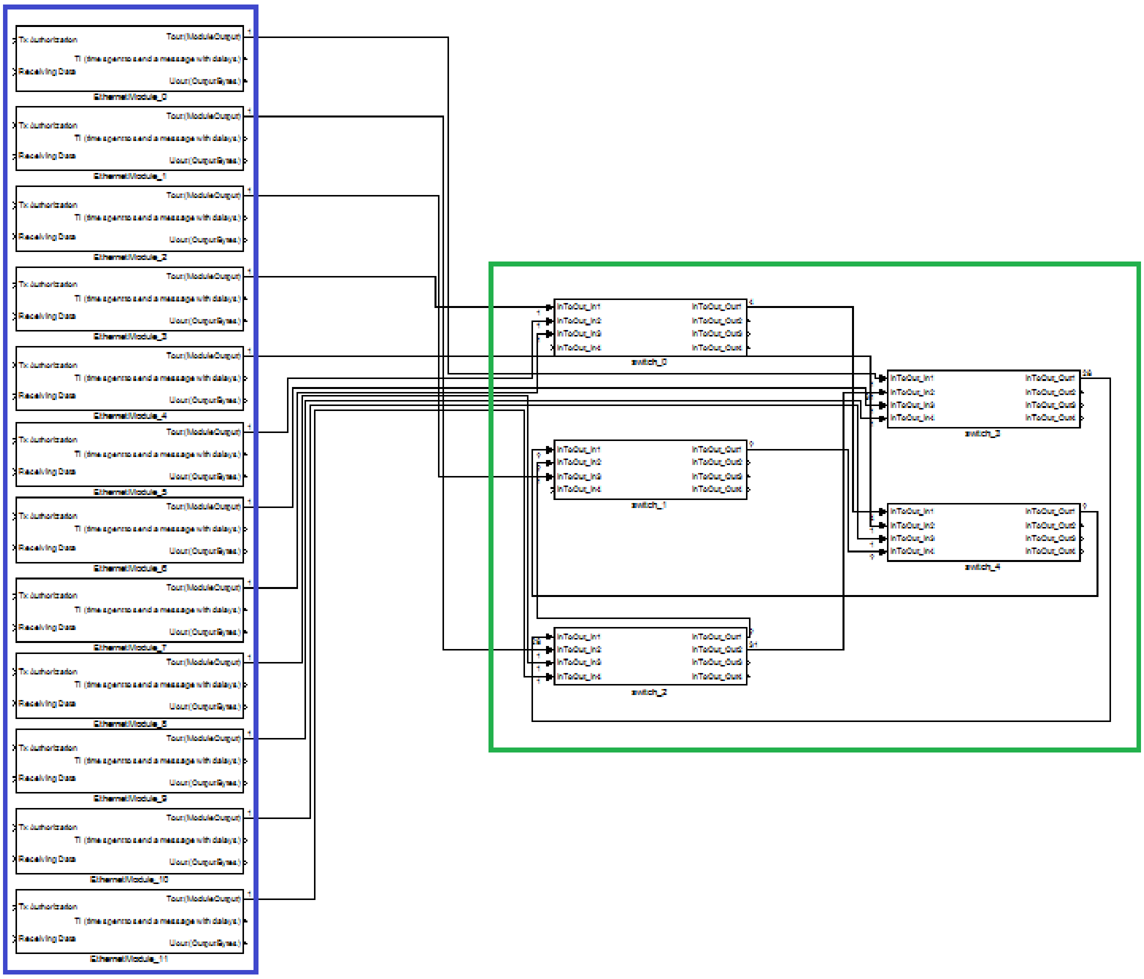


FIGURE 11. The network topology

TABLE 1. The communication matrix among modules

From (Module)	**To (Module)**
5	7
1	4
3	9
0	10
4	7
3	9
4	8
3	1
2	5
2	3
6	8
1	3
7	0
8	6
9	2
10	5

Each possible route is given a score that takes into account the concepts and equations described above.

For the example of Figure 11, the first route gets a better evaluation; therefore, the number of switches used for communication traffic is smaller than the amount used in the second route.

5. **Conclusion.** The proposed manuscript contribution can be verified through the development of a new algorithm to perform the messages scheduling on the bus, in each field module and new temporal equations, applied to industrial Ethernet networks with distributed control, enabling greater rate and volume in the cyclic and acyclic information traffic over the network, due to its time characteristic proposed and then, solving the problems already listed above in 1.1, with the use of a centralized network. Other industrial protocols operate with different distributed control techniques; however, due to the characteristics listed in this paragraph (speed and traffic volume), using the Ethernet standard becomes advantageous.

Moreover, the existence of cyclic and acyclic messages transferred in the Ethernet network is also considered, aiming to solve the problems related to centralized control. To validate the proposed approach, it has implemented a computational tool for the simulation and visualization of the presented concepts.

The results presented in this paper highlight a new concept for distributed control proposed herein, containing 5 (five) field modules and 9 (nine) different control strategies staggered on the bus. The proposed system has been modeled, being possible to show the cyclical communication time.

The paper validation was carried out by developing a computational tool for simulation and visualization of the concepts and furthermore, a real system for actual results comparison.

REFERENCES

[1] V. G. Cerf and R. E. Khan, A protocol for packet network intercommunication, *IEEE Transactions on Communication*, vol.22, no.5, pp.1-13, 1974.

[2] R. M. Metcalf and D. R. Boggs, Ethernet: Distributed packet switching for local computer networks, *IEEE Transactions on Communications*, vol.19, no.7, pp.395-404, 1976.

[3] T. Sauter, The three generations of field-level networks – Evolution and compatibility issues, *IEEE Transactions on Industrial Electronics*, vol.57, no.11, pp.3585-3595, 2010.

[4] P. Brooks, Ethernet/IP – Industrial protocol, *The 8th IEEE International Conference on Emerging Technologies and Factory Automation*, vol.2, pp.505-514, 2001.

[5] M. Felser and T. Sauter, Standardization of industrial ethernet – The next battlefield? *The 6th IEEE International Workshop on Factory Communication Systems*, pp.413-420, 2004.

[6] Internet web site: SENSE electronics company, *Technical Client Report*, www.sense.com.br, 2011.

[7] Ethernet/IP Specification Release 5.0, *Ethernet/IP Normative Elaborated by Open DeviceNet Vendor Association*, www.odva.org, 2011.

[8] A. B. Lugli, M. M. D. Santos and L. R. H. R. Franco, A computer tool to support in design of industrial ethernet, *Journal ISA Transactions*, vol.48, pp.228-236, 2009.

[9] E. M. M. Tovar, *Supporting Real-Time Communications with Standard Factory-Floor Networks*, Ph.D. Thesis, University of Porto, Porto, Portugal, 1999.

[10] M. Zweben and M. S. Fox, *Intelligent Scheduling*, 1st Edition, Morgan Kaufmann Editor, USA, 1994.

[11] A. M. Henriques, *Scheduling Algorithm Communication to the Fieldbus Foundation*, Master Thesis, University of Itajubá, Itajubá, Brazil, 2000.

[12] M. Felser and T. Sauter, The fieldbus war: History or short break between battles? *The 4th IEEE International Workshop on Factory Communication System*, pp.73-80, 2002.

[13] K. C. Lee et al., Worst case communication delay of real-time industrial switched ethernet with multiple levels, *IEEE Transactions on Industrial Electronics*, vol.53, no.5, pp.1669-1676, 2006.

[14] M. Popp and C. Webber, *The Rapid Way to PROFINET*, 1st Edition, 2004.

[15] G. Held, *Ethernet Networks*, 2nd Edition, Wiley Editor, London, England, 2000.

[16] Internet web site: Computer analyze tool, *WireShark – A Network Protocol Analyzer*, www.wireshark.org, 2011.

[17] A. B. Lugli and M. M. D. Santos, *Industrial Networks to Industrial Automation: AS-I, PROFIBUS e PROFINET*, 1st Edition, Érica Editor, São Paulo, Brazil, 2010.

[18] A. B. Lugli and M. M. D. Santos, *Fieldbus Systems to Industrial Automation: DeviceNet, CANOpen, SDS and Ethernet*, 1st Edition, Érica Editor, São Paulo, Brazil, 2009.

[19] L. R. H. R. Franco, *Scheduling Messages to Fieldbus Communication*, Ph.D. Thesis, University of São Paulo, São Paulo, Brazil, 1998.

[20] B. Addad et al., Analytic calculus of response time in networked automation systems, *IEEE Transactions on Automation Science and Engineering*, vol.7, no.4, pp.858-869, 2010.

[21] L. Zhang and Z. Wang, Real-time performance evaluation in hybrid industrial ethernet networks, *Proc. of the 8th IEEE the World Congress on Intelligent Control and Automation*, pp.1842-1845, 2010.

Appendix

Appendix A. Algorithm flowchart to proposed solution.