FIP over Ethernet/IP for Real Time Distributed Systems Implementation

Zakaria SAHRAOUI, Abdenour LABED and Aomar SERIR

Abstract— **This paper presents the realization of a platform for testing and validating distributed real time systems (DRTS), by following a methodology of development. Our main contribution remains the realization of an industrial communication bus (FIP: *Factory Instrumentation Protocol*) implemented on an Ethernet platform. Our work focuses on improving the response time of the bus. For that, we use a deterministic implementation of FIP's services (variables identification, transmission functions,…) by exploiting the TCP/IP stack. The periodic communications are monitored by real time periodic threads run on RTAI kernel.**

Index Terms—***FIP Bus, distributed real time systems, Ethernet, Distributed Grafcet.***

I. INTRODUCTION

Real time and distributed industrial systems development often rests on an appropriate methodology. Their implementation may be conventional by using a programming language or modern by using a fast prototyping tool that involves simulators, code generators and hardware in the loop. So, the design follows a model, architecture and a language appropriate to the applied methodology. The validation step of such systems in particular, requires platforms and middlewares to distribute the controls, the computation or the data. These platforms assure the management of the field buses.

Distributed real time applications must satisfy two conditions to communicate data: determinism and reliability. Conventionally, industrial local area networks or any networks in hostile environment (engine of a vehicle) use field buses such as the controller area network (CAN) and the FIP buses to meet these two requirements. The FIP field bus is a platform offering a configuring interface allowing a station[1] to take place in the FIP network.

Manuscript received March 22, 2010.
Z. Sahraoui is with the computer Science Department of EMP, BP 17 BEB, Algiers, Algeria; (e-mail: z-sahraoui@ gmail.com).
A. Labed is with the computer Science Department, EMP, BP 17 BEB, Algiers, Algeria; (e-mail: a-labed@ hotmail.com).
A. Serir is with the computer Science Department, EMP, BP 17, BEB, Algiers, Algeria; (e-mail: aoser10@ @ gmail.com).

Hence, it can produce and consume periodic or aperiodic variables, send and receive messages with or without connection or assure the arbitration function of the bus. The arbitrator holds the list of variables which is created on configuration. For each variable the producer and its periodicity of use on the bus are defined [1] [2].

Industrial networks tend to exploit the possibilities of the ETHERNET, which has been proven to be well adapted since it has a lower performance/cost.

In this paper we describe a platform that we have developed, to test the DRTS. We have adopted the methodology proposed by M.BENAISSA and A.SERIR in [3]. It is an approach of development based on a descendent hierarchical functional decomposition of a control system. The primitive processes of the system are specified by elementary grafcets that communicate and synchronize using messages. The Grafcets processes may refer to components of the same site or of different sites (distributed). The link between sites is provided by a communication system similar to the FIP field bus with a deterministic, periodic and synchronous behavior. Our contribution concerns a FIP configuration on ETHERNET.

So, we have projected the FIP bus architecture (protocols and services) on its counterpart, the Ethernet communication system together with TCP/IP. This aims at checking the specification of the resulting FIP bus to meet all the mechanisms of FIP : the use of broadcasting in the communication system, acknowledgment of variables identifiers by the different sites and errors verification services (physical, link and application layers of OSI). Consequently, the validation will rest on performances of the real time micro kernel RTAI[2] we have used.

The reminder of the paper is organized as follows: Section II analyses the TCP/IP/Ethernet model. The section III is concerned with the approach for design of the proposed FIP field bus. In section four, we implement an example of a distributed real time application specified according to the methodology. Section V resumes and discusses the results of the tests. We conclude in section VI.

1 Part of the bus that can be, a computer, an automaton, a sensor or an actuator

[2] Open source, preemptible without latency problems of operating system calls and has a known gigue

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. RELATED WORKS

Data exchange between and within field buses requires high rate communications. However, the majority of the field buses such as WorldFip, FF, Profibus, P-Net and CAN suggest insufficient rates (31.25-5Mbps). To overcome this problem, high rate networks as FDDI and ATM were proposed and their capability to support strict time constraints and soft real time has been evaluated by Malcom and Zhao (1995); Song and Simonot (1996) ; Hansson and Sjöd (1997) and Mammeri and Haouam (1997). However, these solutions didn't meet expectations because of their high costs and complexity of implementation [4].

Some improvements have been operated on the Ethernet to help it support communications constrained by the time. So, the Ethernet protocol is modified or a deterministic layer is implemented over the MAC sublayer. One solution consists of using the TDMA strategy. But it has the drawback to waste the time slots of stations whiles they are idle (no transmissions). As examples, we can cite the P-CSMA (*Prioritized CSMA*), RTNET of RTAI micro kernel. The technique PCSMA (*Predictable CSMA*) based on oriented data scheduling and assumes periodic all real time data. Though it avoids time waste, it has an overhead in its off line scheduling. We can also find techniques based on the modification of the binary exponential backoff (BEB) algorithm, like CSMA/DCR (CSMA deterministic collision avoidance) which uses a binary tree research instead of the non deterministic BEB[5]. Indeed, such techniques may support strict as well as soft real time applications by changing the basic structure of the Ethernet. Moreover, adding a deterministic layer upon MAC, may lead to the same result. Among these solutions, we have the *Virtual Time protocol* (VTP), the *Window Protocol* (WP) and the *traffic smoothing* (TS)[4].

Midleware-based protocols of communication have been recently proposed for applications in automation. They are implemented either on TCP, like Modbus TCP and ProfiNet, or on UDP, such as NDDS (*Network Data Delivery Service*) [6] [7]. We end the list of technical solutions by *Avionics Full DupleX switched Ethernet base 100 TX* (AFDX*)*.

III. PROPOSED APPROACH (FIP OVER ETHERNET/IP)

The platform design is the result of Ethernet/TCP/IP model and its counterpart FIP's layers analysis.

A. Analysis of the physical layer

The WorldFIP norm defines three transmission rates : 31,25 kbit/s with a bit transmission time of *Tbit = 32µs*, 1Mbit/s with *Tbit = 1µs* and 2,5Mbit/s with *Tbit = 400ns*[1]. So, the Ethernet rate varies between 10Mbit/s with *Tbit = 51,2 µs* and 10Gbit/s with *Tbit =512ns*[8]. The Manchester II coding is used for Ethernet with 10Mbit/s and for the WorldFIP. For instance, Ethernet 100BaseTx uses the 4B/5B coding which limits the *Tbit* at *8ms*[8]. Note that the effects of noise on the FIP bus are similar as on Ethernet categories which use a short *Tbit* and low modulation speed.

B. Analysis of the data link layer

The first step consists in analyzing the protocols ARP[3] (*address resolution protocol)* and LLC (*logical layer control*) of TCP/IP and the services that may affect the traffic, in order to preserve a deterministic behavior. For our experiments, we used a LAN with one HUB and two PCs on which we installed a linux system provided with a traffic analyzer (Ethereal).

For the Ethernet traffic capture, we noticed that in absence of traffic, three queries may occur on the network: two LLC control queries (initialization), two ARP with the sender address (generated by default every 180s) and active services queries. Furthermore, when we replaced the Hub by a *switch*, we noticed that it also uses the LLC protocol to initialize the network (SSAP query: *Spanning tree* BPDU[4] command with a *forward delay* of 15ns).

To avoid the non determinism of the protocol, it suffices to fix the duration *Base_ Reachable_Time* at an unreachable value. But, wasted time to ensure a deterministic emission or reception (periodic queries of identification, every 180s) can be bounded by *(n+1) Tra*. In this relation, *n* is the ratio between duration of an emission and the period of an ARP query and *Tra* is the transmission time of an ARP query.

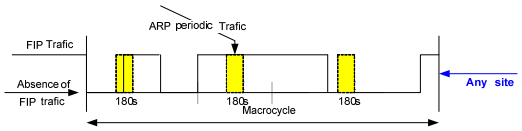

Figure 1:Computation of the delay for the periodic traffic

C. Benefit of the data link layer

The FIP frame format on the link level (Fig. 2) involves a control byte to code the frame's type (ID_DAT, RP_DAT, ID_RQI, RP_ACK,…), data bytes (128 bytes or 256 octets) and two bytes allowing a receiver to check the integrity of the received frame[1].

PRE	DDT	Control	Data	CRC	DDT
8 bits	8 bits	1 octet	X octets	2 octets	8 bits

Figure 2: Link layer frames format of the WorldFIP *(CENELEC EN61158-2 norm of FIP)[1]*

Source and destination MAC addresses of Ethernet frames [5] (Figs. 3 & 4) have no role in the specification, since sites identification in FIP bus has no interest at this level.

[3] ARP : protocol of layer three which makes the correspondence between Internet logical addresses and MAC addresses.
[4] Used by *switches* and routers to avoid loops on a WAN.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

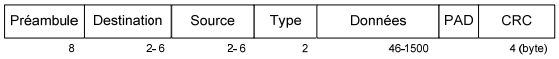

Figure 3: Ethernet II frame

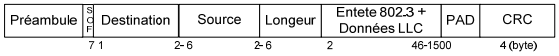

Figure 4: The Ethernet IEEE 802.3 frame

Using Ethernet CSMA/CD protocol transmission errors are not detected through the absence of acknowledgment, but through interference. In FIP implementation over ETHERNET, the temporization is assured implicitly. However the frames (identifier, variables response, query response,…) are processed at transport and application layers. Error control is required by the CRC for both FIP and Ethernet using the same code.

D. Benefit of the network layer (IP) and the broadcasting principle

In the OSI standard, the TCP/IP protocol offers routing operations. So, interconnection between any pair of machines is possible. But in FIP network, sites identification is implicitly provided by the identifiers of variables to be transmitted. Recall that client sites of FIP system are synchronized only by variables identifiers and IP address will not give information on variables identifier. Consequently, in our design, sites participating to the exchange of variable are implicitly identified as producer of this variable via the broadcasting principle. The use of HUBs offers implicitly the broadcasting possibility. But, if we use *switches*, an appropriate configuration is necessary (configure ports on promiscuous mode).

E. Benefit of the UDP layer

UDP protocol has been the unique solution for many tools of real time applications implementation. UDP datagrammes nature is ideal for sending fragments of data generated by such applications. It is particularly selected because of the speed of communication between its clients. It uses a simplified structure of the header, which restricts to the fields: source port, destination port, datagramme lenght and checksum(Figure 5).

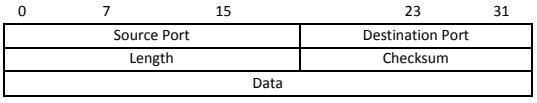

Figure 5: Format of UDP datagramme or message.

The checksum of the header is computed as for IP packets, a checksum at 0 indicates that it is not used.

F. Adopted architecture for the transport level

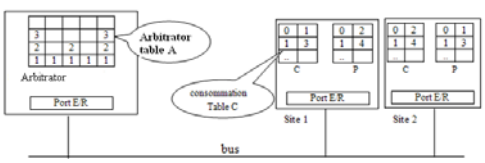

Figure 6: Architecture of the adopted Transport level.

Architecture of figure 6 shows the solution we have chosen among much other architecture. Use of tables *P* and *C* to structure producer and consumer variables, simplifies managing variables at the processing step. Different port numbers are used for the arbitrator and the producer-consumer sites, to separate messages intended to identify variables and those which contain variables. Using a unique port instead of many at each producer/consumer sites, allows synchronous design of production and consumption functions.

A producer site receives identifiers ID_DAT of the variables to be produced. The consumer detects the arrival of these frames in order to enable an internal temporizer. If this temporizer expires, the station considers the next frame only if it has the emitting port of the arbitrator.

Recall that messages exchange on the FIP bus is done in point to point or in multipoint on the same segment. Two addresses of 24 bits (source & destination) allow coding the number of the segment of the application entity and its address on this segment. Hence, IP addressing may be used to perform these transactions.

G. Maximal transferring time (critical time)

Real time and distributed applications impose temporal constraints tasks achievement; these constraints will have direct impact on the exchanged messages between tasks located on different processors. In real time application, tasks may have or not temporal constraints as well as the exchanged messages between them.

As indicated on figure 7, the transferring time of message is composed of several intermediate times which are summarized in table I. Note *Dt* the duration of a transaction, then:

$$Dt = Tta + 2\,Tst + 2\,Tsm + 2\,Tp + 2\,Trm + 2\,Trt + 2\,Trp + Ttp \ldots\ldots\ldots (1)$$

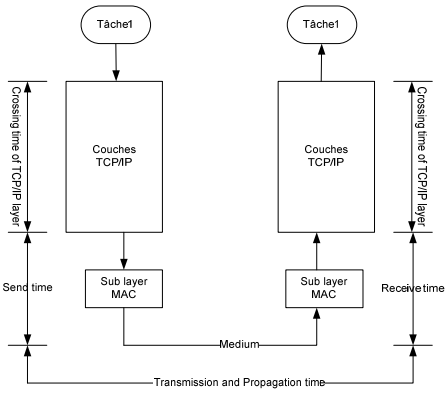

Figure 7: Ttransmission times.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I. NOTATION USED FOR TRANSMISSION TIMES.

Transmis. of identifiers	Notation	Transm. of variables	Not.
Identifier sending time (FIP Arbitrator task)	Tta	variable sending time (FIP production Task)	Ttp
Latest *Sendto(socket,UDP…)* time	Tst	Latest *Sendto(socket,UDP…)* time	Tst
MAC emission time	Tsm	MAC emission time	Tsm
Transmission time on the medium	Tp	Transmission time on the medium	Tp
MAC reception time	Trm	MAC reception time	Trm
Time *Recvfrm(Socket, UDP…)*	Trt	Time *Recvfrm(Socket, UDP…)*	Trt
Acknowledgement Time (FIP producer task)	Trp	Checking time (Tâche Arbitre FIP)	Trp

Given the fact that in our protocol, service is carried out by sources of indeterminism, which are all periodic, the maximal time of periodic transaction is computed as:

$$Dt \leq Tta + 2Tst + 2Tsm + 2Tp + 2Trm + 2Trt + 2Trp + Ttp + (n+1)Tra \quad \ldots\ldots\ldots (8)$$

IV. IMPLEMENTATION OF THE COMMUNICATION SYSTEM

A. Hardware and software of the platform

Each component materialized in our case by PC provided with an Ethernet network interface controller 100BaseT[5] is considered as a site. On each site the RTAI system is installed. We have used four: three are Pentium 4 with CPU of 2.39 GHz, the fourth is a Pentium 2 with CPU of 233 MHz. The first machine plays the role of arbitrator, the second and the fourth the role of producers-consumers (site 1 and site 2). The third has a monitoring role by analyzing trafic using the Ethereal tool.

B. Our arbitration function

Variables identifiers are scheduled on a macrocycle as follows:

- associate a task to each identifier;
- each task must be triggered periodically in the macrocycle at a precise time of the elementary cycle to broadcast the identifier;
- arbitrator executes tasks thanks to a preemptive scheduler with priority. A task is elected by scheduler only if all its preceding tasks have been achieved;
- remaining time after execution of all the tasks in a microcycle, will be used for aperiodic exchange ;
- awakening date of a periodic task *i* is computed by taking into account the total transfer time of all the previous transactions of a microcycle.

[5] Network interface controller: we limit the study to Ethernet 10 and 100 Mbps.

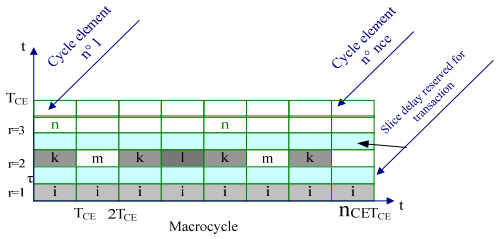

Figure 8: Modified macrocycle.

C. The Producer-Consumer function

We will specify production and consumption tasks of a site. The first task to be executed is the production function, because, the client site have first of all to wait for an eventual identifier of a variable using the primitive *recvfrom(int s, void *buf, int len, unsigned int flags, struct sock-addr *from, sockelen_t *fromlen)*. Then, the producer task scrutinizes its table to check if it is concerned by the variable associated to this identifier in which case it broadcasts the variable. If the site is not the producer, the same task scrutinize it consumption table to check if it has to receive the variable on the same port using the same primitive. On the other hand, in the consumption processing, the task enables an internal temporizer to confirm the frame loss at expiration of this temporizer and assure the global order of the system.

D. Use of real time FIFO mechanism

Technically, it was not possible to compile a new Ethernet network driver over RTAI. So, ETHERNET of Linux system is used via the mechanism of real time queues (RT_FIFO), to communicate between ordinary processes and RTAI real time processes. The arbitrator creates two RT_FIFOs for its services; the reason is that, the primitive *(rtf_get)* used to read variables will use another mechanism of asynchronous nature. This primitive has been put in function that we have called monitoring.

E. Monitoring function

This function focuses on the variables exchange via real time queues and computes the transaction time. It is automatically enabled by the arrival of a variable in the queue (linux process has inserted the variable in the queue). This mechanism is assured by *rtf_create_handler(fifo, supervision)* primitive. Notice that contrary to the FIP bus variables, the RT_FIFOs' buffer has no refreshment and promptitude problem.

F. Schedulability

In our implementation we used an arbitrator which involves a set of RTAI periodic tasks and another function for the producer-consumer site. The latter is sequentially executed and respects the order of a FIP transaction. The fact that arbitrator tasks are periodic makes it possible to apply the schedulability criterion of formula (3) and to compute maximal times of transactions execution.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\sum_{i=1}^{i=n} DT_i / PT_i \leq 1 \text{ .....(3)}$$

DTi : i^{th} FIP periodic transaction time,
PT_i : i^{th} FIP period transaction,

V. A CASE STUDY

In this example, the application is composed of two distributed grafcets in the communication system (fig. 9)

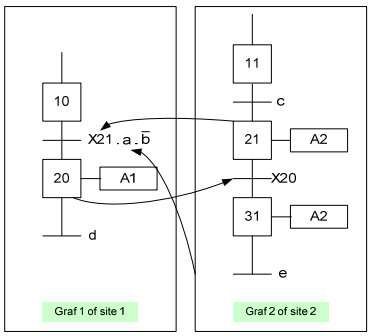

Figure 9: Example of distributed grafcet

Communication between the two grafcets requires transmission of input a(m1) and state X21(m2) of site 2 to site 1 and transmission of state X20(m3) from site 1 to site 2, for each period of the macrocycle. Message m1 is transmitted during the first elementary cycle. Messages m2 and m3 are transmitted during the second in the ordre m2, m3.

To assure a coherent functioning of the arbitrator, we have bounded the time of a transaction, and consequently, the time to be added to the periods of variables. So, we compute the values resulting from substractions between the clock value read after every sending and the corresponding value of the clock sent by the monitoring function. Then, we take as upper bound the maximum of the obtained results.

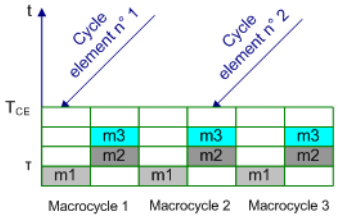

Figure 10: Arbitrator table

A. Experimental results

In the test step, we have considered the parameters : time of transactions and the duration of production function, to compare our solution to the original FIP.

Since we have initialized the period, in timer tick, (periodic mode) to 119 ticks or 100000 ns, a time value in tick of the clock is converted to nanoseconds by multiplying it by 840.336.

We give bellow two sets of results (Fig. 11), corresponding respectively to a setup with a 10BaseT HUB and with a 100BaseT *switch*. Notice tha we have used cables of UTP5category.

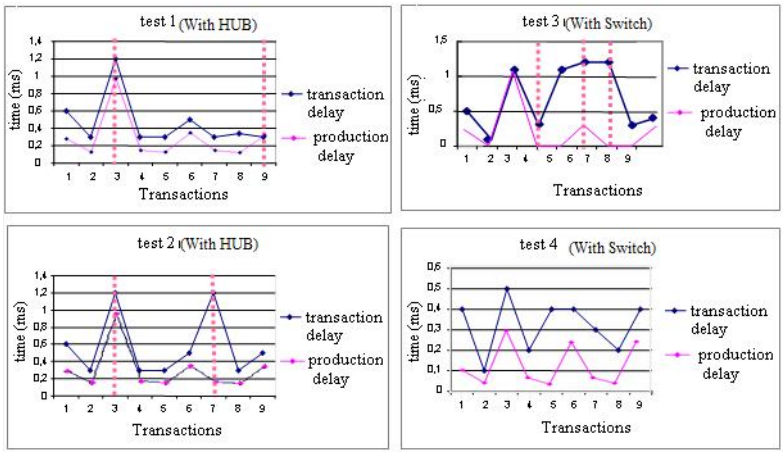

Figure 11: Results Interprétation

We have estimated the time used by a producer to produce the frame response of variable m_i and the associated propagation time.

B. Discussions

Table II, gives an idea about some mesured times.

TABLE II. TRANSACTIONS TIME EN MILLISECONDES

	Test 1	Test 2	Test 3	Test 4
Average	0.460	0.600	0.700	0.322
Maximum	1.2	1.2	1.2	0.5

Values sent by the identifier sending task and those sent by the monitoring function are given in tick of clock. The results of substractions between the durations are converted into seconds.

Minimum	0.3	0.3	0.1	0.1

We can notice that transaction times may be lowered using a *switch*.

The maximal values are almost equivalent for all the tests and varie between 0.5 ms and 1.2 ms. These results may be interpreted by the fact the production times are often less than the half of

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

transaction times, which explains the slowness and inderminism of emission and reception function of linux arbitrator.

For example, let us take the value 1.2 ms which represents time of the seventh transaction of test 3, and the value 0.047 ms the time of its production. We notice that delay is due to emission and reception functions of the arbitrator (Fig.11).

Another exampl, concerns the maximal value obtained in test 1, it corresponds to the the transaction of variable m3. This value of 0,979 ms gives the production time of the variable by the slowest machine (site 2).

C. *Speed of variables scrutation*

Note that FIP Network at 2.5 Mbps, with a reversal time of 10 ms, in an element cycle of 20ms, we can scrutinize:

TABLE III. ECHANTILLON SUR LA RAPIDITE DU FIP ORIGINAL [3]

320 variables	1 byte	181 variables	16 bytes
304 variables	2 bytes	123 variables	32 bytes
277 variables	4 bytes	75 variables	64 bytes
235 variables	8 bytes	42 variables	128 bytes

From table II, we can get the interval of variation for the scan speed of the arbitrator (Fig.12). We can notice that in the case of the *switch*, the scan speed may reach 200 variables per 20ms. However, for the FIP this speed is reached if the size of variables passes from 16 to 8 bytes.

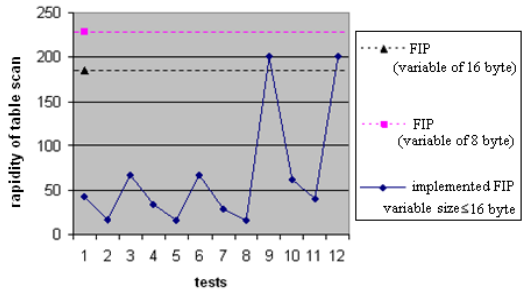

Figure 12: Comparison of scrutation speeds.

D. *Useful bit rate*

The useful bit rate is the ratio of the effective information and the duration of a transaction. For variables of length *L: 1 <L< 16* bytes, a MAC frame has always a size of 64 bytes. But, if the variables have a length *L* greater than 16 bytes, the size of the MAC frame will vary between 64 and 1500 bytes. Nevertheless, transmitted information is segmented into frames of 1500 bytes if the variables size exceeds 1453 bytes.

E. *Efficiency*

We now compute the efficiency of our solution as the ratio between emission time of effective information and the duration of the transaction. It is equivalent to the ratio between the useful bit rate and the transmission rate. Table 19 compares the FIP's [2] efficiency and that of our implementation. To complete the table, we deduce FIP transaction time from the ratio : length of useful information and useful bit rate.

To compute the useful bit rate, we take the global minimum of all the durations (tests of previous example). The tests on the implemented FIP, with a variable of 32 bytes gave the same minimums.

TABLE IV. COMPARISON BETWEEN THE STANDARD FIP AND IMPLEMENTED FIP

	Length of useful information (Byte)	Length of transmitted information (Byte)	Transaction time (ms)	Useful bit rate (Kbps)	Efficiency
FIP	4	19 ou (16+4)	0,072	444,44	17,77 %
	8	23	0,084799	754,72	19,27 %
	16	31	0,013800	1159,42	32,32 %
	32	47	0,020200	1584,16	48,85 %
	64	79	0,033000	1939,39	65,64 %
	128	143	0,058600	2184,30	79,25 %
Implemented FIP	4	64	0,099999	5120	05,12 %
	8	64	0,099999	5120	05,12 %
	16	64	0,099999	5120	05,12 %
	32	79 ou (47+32)	0,099999	6320	06,32 %

This table gives an idea about the margin that we have on the size of data that we can transmit in a transaction. Hence, efficiency is increased if we add other services (aperiodic variable exchange).

VI. CONCLUSIONS

We have compared the results obtained for the distributed grafcet and the standard FIP on a practical example. The comparison was concerned with the scan speed of arbitration table and the computation of the communication system efficiency in its cyclic part. Results obtained using switched Ethernet (*switches*) show that arbitrator of the FIP we have implemented can scrutinize its arbitration table faster than the standard FIP.

The implemented platform confirms the goodness of the distributed grafcet model. It constitutes by itself a new design for implementation of distributed real time systems which can be qualified as distributed systems for field data base management.

REFERENCES

[1] WORLDFIP tools FIPDESIGNER hlp technologies. L M 2 - C N F - 2 - 0 0 1 – D, 12 Jul. 2000.

[2] WorldFip Protocole. European standard, En 50170, in http://www.WorldFIP.org, 1999.

[3] Mohamed BENAISSA, '*GRAFCET based Formalism for design of distributed real time systems'*. Master Thesis, EMP Bordj-El-Bahri, Algiers, Algeria, 2004 (in french).

[4] Zhi WANG, Ye_qiong SONG, Ji_ming CHEN You_Xian SUN, '*real time characteristics of Ethernet and its improvement'*. Proceed. Of the World congress on Intelligent Control and Automation, june 2001.

[5] Guy PUJOLLE, Networks. Eyrolles Second edition, 1997 (in french).

[6] Chitra VENKATRAMANI, Tzi-cker CHIUEH, '*Supporting real –time traffic on Ethernet'. 1052-8725/94 IEEE.*

[7] Ondrej DOLÉJS, Zdenek HANZALÉK, '*Simulation of Ethernet for Real Time Applications'.* IEEE, ICIT - Maribor, Slovenia, 2003.

[8] Telecommunication and Networks, Claude Sevin, Dunod 2006 (in french).

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)