Journal of Engineering and Computer Sciences,
Qassim University, Vol. 3, No. 2, pp. 163-183 (July 2010/Rajab 1431H)

A Real Time Network Communication Interface: Implementation and Performance Evaluation

Salem Nasri
College of Computer
Qassim University
salemnasri@yahoo.fr, nsary@qu.edu.sa

(Received 23/6/2009, accepted for publication 3/3/2010)

Abstract. The architecture and the performance of a network interconnecting interfaces are of great importance. Indeed, the majority of these interfaces function have time-critical constraints that must be considered when designing such a network. Performance can be evaluated in different ways using parameters measurement: end to end delay, time of request-response protocols, data losing or network throughput etc. In case of industrial environments time is a critical parameter, and the performance can be assessed in terms of estimated time of the request-response services.

In this paper, an architecture based on the OSI reference model is presented, along with service elements and protocols, which match the requirements for these constraints. This architecture is an application layer protocol based on ATM (Asynchronous Transfer Mode) and MMS (Manufacturing Message Specification). We study and design its software and hardware implementation. Then we evaluate the performances of this implementation from external and internal points of view. To understand how time is spent in different levels of the communication system, we study the transmission delay needed by the execution of the MMS protocol layers over ATM protocol. To increase the performance of the communication system we propose to reduce the number of layers in a Mini-MAP architecture, by offloading high overhead MMS functions to a hardware co-processor. Software parts of the architecture, especially the buffer management have been carefully implemented.

The target architecture is based on three design choices:

1. High-speed communication fiber-based providing 155 up to 622 Mbps ATM network and supporting quality of service (QoS) parameters.
2. Offloading of high overhead functions of the MMS protocol to a hardware co-processor,
3. Implementation of software parts of the MMS protocol.

***Keywords*:** ATM, MMS, Industrial Networks, MAP, Protocols.

1. Introduction

a) *MMS overview*

MMS is an OSI application layer protocol that enables remote applications (called clients) to control and supervise various heterogeneous industrial devices (called servers). MMS is a part of the Manufacturing Automation Protocol (MAP) architecture [1, 2, 3]. It defines only two aspects of communication in an industrial environment. The first one is the concept of a Virtual Manufacturing Device (VMD), which essentially presents an abstract view of a physical device and the second one covers objects, services and protocols used to support communication between such abstract devices [2, 3, 4]. The VMD hides the complexity of the real industrial device and provides a common understanding of all such devices. MMS defines a large set of services and object classes [4, 5].

MMS is widely accepted for open communication between heterogeneous devices in many areas, not even limited to manufacturing environment.

Mini-MAP is a version of MAP consisting of only physical, link, & application layers intended for lower-cost process-control networks. A Mini-MAP device with a token can request a response from an addressed device; unlike a standard MAP protocol, the addressed Mini-MAP device need not wait for the token to respond.

b) *The ATM Plant Control Network*

ATM networks are getting a large place in the industrial communication networks. Two key techniques are used for ATM networks: asynchronous multiplexing and fast packet switching. Asynchronous multiplexing enhances the efficiency of a network to transport varying bit-rate traffic since many users can dynamically share a transmission link. Fast packet switching reduces the message-transmission delay by implementing switching functions in hardware [6, 7].

ATM is a technology used to provide integrated services for high-speed digital communication networks. Its ability to support high bandwidth, high reliability and guaranteed quality of service communication makes it a candidate for the plant control networks [8, 9].

This paper contains the following sections: section two describes the MMS/ATM communication system architecture in which four layers are specified. The implementation of the MMS/ATM services is detailed in section three. Section four focuses on the performance evaluation from an external and an internal point of view. The analysis of the proposed solution is presented in section five. The paper is ended by tentative list of improvements and conclusions.

2. Mms/Atm Communication System Architecture

The goal of the MMS architecture is to provide a new communication system in an industrial environment. To achieve this goal, we have chosen to use ATM and to reduce the number of layers by interfacing the MMS application layer directly to the ATM adaptation layer [8, 10]. MMS/ATM architecture consists of four layers

(Fig.1):

1. MMS application layer,
2. ATM adaptation layer (AAL),
3. ATM layer
4. Physical layer.

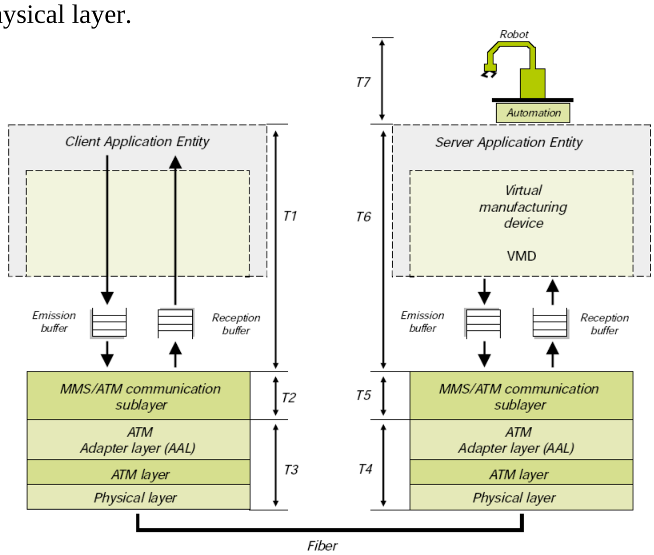

Fig. (1). MMS/ATM Communication System Architecture.

The three lower layers conform to the ATM reference model and are implemented on the communication system. The MMS application layer is implemented as software on a host processor.

a) The MMS application layer

The MMS application layer is split into two parts: MMS Application Entities, MAE (client and server) and MMS/ATM communication sub-layer (Fig.1). The Client Application Entity presents the MMS primitives to applications and communicates with the Server Application Entity (VMD) on a remote device via the MMS/ATM communication sub-layer. The application entities communicate with the lower layers by means of two buffers: the Emission Buffer and the Reception Buffer [11, 12].

We describe below the lower communication layers implemented on the ATM communication system.

b) The Lower Communication Layers

An important sub-layer in the ATM stack is the ATM adaptation layer (AAL). The function of the AAL is to convert and map information from the higher layers into the ATM layer (and vice versa). AAL functions are arranged into two sub-layers: the Segmentation And Reassembly (SAR) sub-layer and the

Convergence Sublayer (CS) [13].

The AAL layer provides two access interfaces: the user access interface and the signaling access interface. The signaling access interface allows an application entity to initialize the ATM layers as well as to establish, release, and abort a connection. The user access interface provides primitives for opening and closing ATM ports that are used to send or receive data.

The ATM layer provides a transparent cell switching service to the AAL layer. Our system contains an embedded 4x4 ports ATM switch. It is the principal component of the communication system, based on the concepts of VCI (Virtual Channel Identifier) and VPI (Virtual Path Identifier) in the cell header. It switches incoming cells to the next ATM switch [13, 14].

c) The MMS/ATM Communication Sublayer Specification

This section presents the main contribution of the MMS protocol implementation directly on the top of the AAL layer. Our contribution defines, essentially, the interface between the MMS and the AAL layer. This interface is called MMS/ATM Communication Sublayer and its architecture is presented in fig.(2). This sub-layer allows the adaptation between MMS Application Entities (MAEs) and the AAL layer.

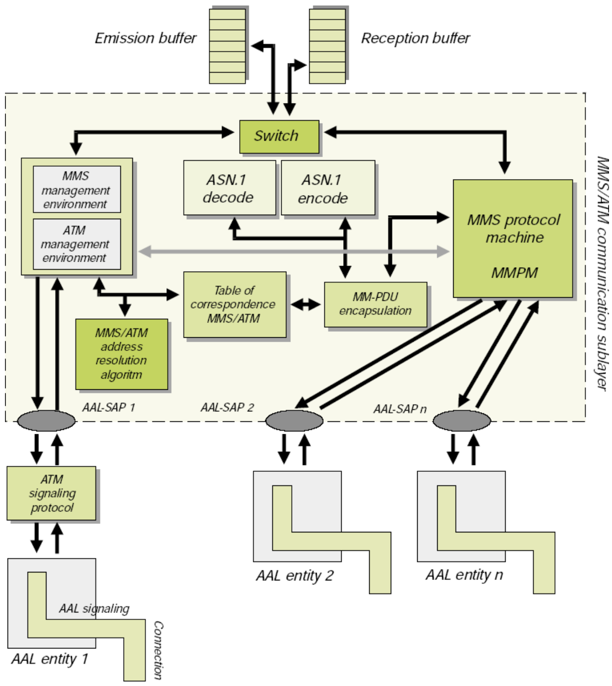

Fig. (2). MMS/ATM Communication Sublayer Specification.

The main functions performed in the MMS/ATM communication sub-layer are described in the following subsections.

i.) *The MMS Protocol Machine (MMPM)*

The MMPM controls the exchanges between the communicating entities. The MMS protocol is executed by an instance of the MMPM state machine, which must ensure the link between MAEs and the AAL layer. This MMPM converts the local syntax of MMS primitives and data to the transfer syntax (encode/decode functions). It is different from previously proposed MMS implementations.

The client MMPM state machine performs two actions. *Action1* is the search of the MMS association identifier that corresponds to the AAL connection identifier and the encapsulation of the MMS-PDU to an AAL-PDU. However, *Action2* is the search of the AAL connection identifier that is associated to the MMS association identifier and the encapsulation of an AAL-PDU to an MMS-PDU.

ii.) The *MMS-PDU encapsulation*

There are one emission and one reception buffer for all associations in the architecture. Each application entity must distinguish its MMS-PDU at the reception. At emission, the MMPM needs to know the destination application entity. Hence, the MMS-PDU must be encapsulated. This operation of encapsulation is necessary to route the MMS-PDU through the corresponding AAL connection.

Emission: Each application entity must specify the value of its *AssoRef* and the called *AEid* before adding its MMS service parameters to the emission buffer. When the MMPM receives these *AssoRef/AEid* values as a header for an MMS-PDU, it looks in the MMS/ATM correspondence table for the AAL-SAP that corresponds to this *AssoRef*. Then it encapsulates this MMS-PDU into an AAL-PDU, and sends it to the called node through the network.

Reception: At the called node, (works also for the calling node with a confirmation primitive), when the MMPM receives the MMS-PDU at an AAL-SAP, it looks for the corresponding *AssoRef/AEid* parameter in the MMS/ATM table. Then it sends the *AssoRef/AEid* value with the result of the decode procedure of the received MMS-PDU to the reception buffer.

iii.) *The MMS/ATM address mapping*

An association (which is a connection in the application level) supports the communication between two *MAEs*. If an *MAE* wants to create an association with another *MAE* through a network, the calling entity must give the address of the called entity to the lower layers. The lower layers used in our communication architecture define only the ATM addresses that are specific to each AAL connection.

The MMS association is characterized by the *AssoRef/AEid* parameter. This parameter identifies a local Association Reference (*AssoRef*) and a destination Application Entity identifier (*AEid*). The AAL connection is identified by an AAL Service Access Point (*AAL-SAP*) and a *VCI/VPI* value [15].

We suppose that each *AAL-SAP* corresponds to only one MMS association. After opening an association, a table of correspondence between MMS associations and AAL connections (Fig.2) is updated. This table maps each *AEid* to an AAL-SAP and VCI/VPI.

iv.) *The MMS management environment state machine (EMPM)*

This function controls the right sequencing of service primitives and contains an MMS management environment entity to allow the establishment of an MMS environment. It provides only point-to-point communication primitives at the MMS level associations.

This state machine passes the parameters to the ATM Environment Management (AEM). These parameters are necessary to initialize an AAL connection that will be used by the EMPM state machine to initialize the MMS environment. When an ATM environment is setup correctly, the EMPM can use this ATM environment to send primitives to the MMS environment management service (like Initiate, Conclude, etc).

d) The MMS/ATM OSI Functionalities.

The MMS/ATM links directly the application layer and the ATM adapter layer (AAL) by-passing the network, transport, presentation and session layers. Below, we study the consequences of this choice [2, 16].

At the current state of networking technology, the probability of transmission errors is negligible. So we can by-pass the transport and session layers that take care of error-free transmission and recovery. Because of the absence of the transport and network layers, the flow control must be implemented by using an MMS parameter that imposes a maximum number of outstanding requests for services. The ATM technology provides some internetworking functionalities of the network layer. It switches ATM cells from source to destination. Also, we have defined a table of correspondence between MMS associations and AAL connections, which achieves the function of address mapping, performed in a classic network layer.

3. Implementation

For this realization we have adapted a hybrid hardware/software implementation which is composed of:

Software system: Consists of a program, which realizes MMS communication tasks.

Hardware System: Consists of an ATM communication system that implements the first three layers of the MMS/ATM system. We have chosen the ATM Adaptation Layer type 5 (AAL5) because this light weight protocol is sufficient for classical data and most types of real-time traffic. The AAL layer is implemented in hardware.

In the following section, we discuss, in first, the size of the PDU buffer. Then we give a method to manage the emission/reception buffer. Then, we present the ASN.1 encoding/decoding function used in the implementation and we describe our ATM communication system. Finally, we define the communication interface between the ATM and the MMS application layers.

a) The PDU Buffer Size

The choice of the PDU (Protocol Data Units) buffer size depends on the size of the messages to be transferred. In industrial environments the messages are relatively short. In the case of the Mini-MAP architecture, the maximum length of PDUs is 1024 bytes [1]. This restriction is not imposed by the MAC layer, which can transmit up to 8 Kbytes, but it is due to performance considerations, namely bounds on token rotation times [17]. This problem does not apply in the case of our architecture since we use a high speed ATM transmission. The only limitation is the size of the segmentation and reassembly buffer managed by the AAL component. The length of this buffer is fixed to 128 Kbytes. For example, 4 Kbytes is the average message length when we are operating with 32 bit connections.

For our MMS/ATM implementation the size of the mono-block PDU buffer is 1024 bytes. This choice is motivated by allowing a lower cost at encoding/decoding time [17].

b) Emission/Reception Buffer Management

Buffer management is critical task in protocol implementation. There are two strategies for the choice of the architecture of an emission/reception buffer: mono-block or multi-blocks. The first approach is to allocate one buffer for all associations created by the MMS application entity. The disadvantage of this first approach is the wasting of a considerable amount of memory when we have a little number of associations. The second approach is to allocate small buffers for each association. This approach reduces the amount of unused memory but has the disadvantage of being more complex to manage. For simplicity reasons, we have chosen to implement the first approach.

In this way it is necessary, in our opinion, to give a mechanism that must control the buffer emission and reception (Fig.3). The communication between two application entities, making use of MMS, is made on the basis of associations. When an MMS association is established, the maximum number of pending MMS requests is negotiated between the AEs. There are neither transport nor network layers in our MMS/ATM architecture. This is why we have chosen to implement the flow control by using a new MMS parameter that controls the padding of the emission and reception buffer.

This MMS parameter is specific to client and server application processes. When an MMS client application sends a request service to an MMS server application, the client must indicate to the server the state of his reception buffer (empty, overflow) by using a *Buffc* parameter. Inversely, an MMS server application must signal to an MMS client application the state of his reception buffer (empty, overflow) by using a *Buffs* parameter.

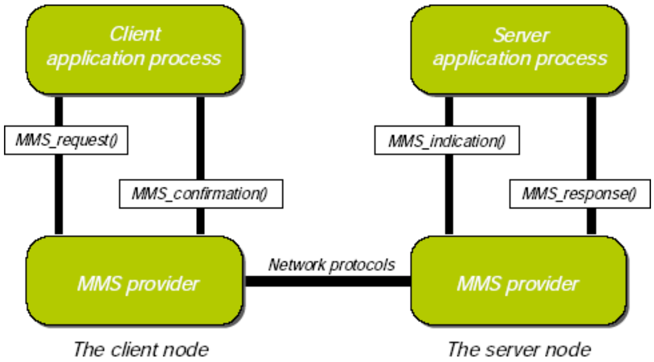

Fig. (3). MMS Communication Sequences.

c) ASN.1 Encoder and Decoder

In Mini-MAP architecture, single transfer syntax may be used to encode and decode MMS PDUs. We use the same syntax for the MMS/ATM architecture. The ASN.1 basic encoding rules have numerous options [18]. The main option concerns the way in which the length field of a composed type is encoded. There are two possible formats: the definite format and the indefinite one. The definite format of encoding is much slower than the indefinite one [17]. This is why we have decided to encode in indefinite format.

There are, essentially, two approaches for parsing application PDUs (decoder) described in ASN.1. With the table-driven approach, the syntax of PDUs is reflected in tables which are used by a protocol independent of the parsing routine. On the other hand, in the code-driven approach, the syntax of PDUs is reflected in the control structure of the parser, which is protocol dependent. We have chosen to work with code-driven parsers because of their speed.

d) ATM Interface Communication

The ATM system (Fig.4) offers a communication service interface for the workstation and incorporates a 4x4-switching element. The chosen physical interface operates at SDH STM-1 (155.22 up to 622 Mbps). The switch and the header translator are the principal components of the communication system. The switch routes incoming cells to the workstation (or to the network) depending on the VCI and VPI values in the *Cellheader*. The routing tables are set or altered dynamically by the signaling management software when connections are opened or closed.

The ATM Adaptation Layer type 5 (AAL5) chosen is hardware implemented. The communication between the AAL chip and the microprocessor is done by means of a shared memory and programmable interrupts (on a message or on a set of cells). A double port memory (DPRAM) has been chosen as the segmentation and the reassembly buffer. This avoids huge traffic on the system's 32-bits bus. If traditional memory (one access) is used, all data would have been carried twice on the bus, which would require high bandwidth on the system bus. Nevertheless, DPRAM modules have very large sizes and low capacities. In order to concurrently process several connections, it is necessary to have about one megabyte of capacity or more. Therefore the chosen solution consists of a small DPRAM (128 KB) as a temporary buffer. This ATM system is connected to the high speed PCI bus of the workstation.

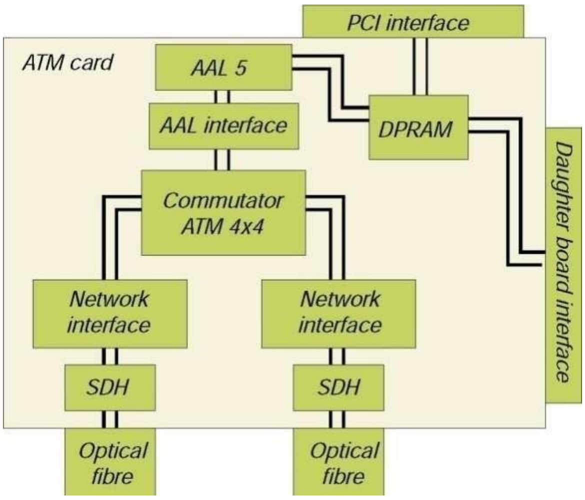

Fig. (4). Communication system.

e) Communication between ATM board and MMS application layer

The MMS Protocol Machine (MMPM) contains a set of application primitives to access the ATM driver (Fig.5). These primitives are of two types: signaling primitives and user primitives. The first type is used by the MMPM to execute signaling functions such as initialization or termination of a connection. The second type allows MMPM to send and receive data.

- The user primitives allow to:
  - release an ATM port when the transfer is terminated (*Close*),
  - receive data from an ATM port (*Recv*),
  - send data through an ATM port (*Send*),
  - create an ATM port attached to an access point (*Attach*).
- The signaling primitives allow to:
  - close a connection after the reception of an application request (end of treatment) or forced by the network (*Kill*),
  - deactivate the ATM board (*Close*),
  - initiate the ATM board (*Init*),
  - open an ATM connection (*Connect*).

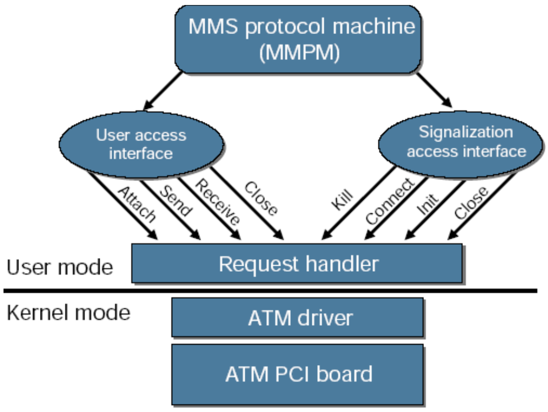

Fig. (5). ATM Board Interface.

4. Performance Evaluation

The goal of the performance evaluation is to determine the improvement provided by the MMS-ATM architecture. The measurements are limited to the MMS request-responses time between two MMS-ATM nodes. The MMS standard is composed of a set of services and objects. Our purpose is not to implement all the MMS services but to design a communication system that will provide an efficient communication in an MMS environment. Our implementation provides only the execution of a subset of MMS services defined by the standard. These services are *Initiate, Start, Write, Read, Download,* and *Conclude.*

a) MMS services description

The MMS services implemented in our system can be described as follows:

- *Read service*: the client uses the read service in order to request that a server returns the values of one or more variables.
- *Write service*: the client writes variables at the server. In the service call a specification of the variables with their corresponding values is conveyed.
- *Initiate service:* The Initiate service is used to establish a communication between two MMS-users. The Initiate service negotiates the conditions for a communication with the MMS-partner.
- *Conclude service:* The Conclude service is used to terminate a communication orderly.
- *Start service:* This service starts the Program Invocation. The Program Invocation must be in the *Idle* state for this service to be valid.
- *Stop service:* The Stop service causes a transition from the *RUNNING* to the *STOPPED* state of the Program Invocation. The progress of the Program Invocation will be stopped.

- *Download service:* The Download service is composed of three services. The first service is the Download service and is used by the client to ask the server to begin loading. The second service is the Download Segment service that is used by the server to ask the client to start the download after receiving the Initiate Download service. Finally, after the server has received all the segments, a *Terminate Download service* is sent to the client.

In the case of a confirmed service, the measured time *T* represents the duration between the transmission of a request and the reception of the confirmation by the client program (Fig.6). In case of an unconfirmed service, the transmission delay is the time between the transmission of a request by the client program and the reception of the indication primitive by the server program.

Some MMS services are needed to manage the ATM environment. Primitives *C* in Fig.(6) represent these MMS services which are:

- *Link :* This service is used by an *MAE* to establish an MMS environment. This *MAE* must send to the *MEM* all parameters necessary to initiate an MMS association and ATM connection.
- *ATMinit:* This service defines the ATM signaling protocol parameters necessary to establish an AAL connection. It is used to initiate an ATM environment.
- *ATMclose:* This service is used by a *MEM* service to close an AAL connection.

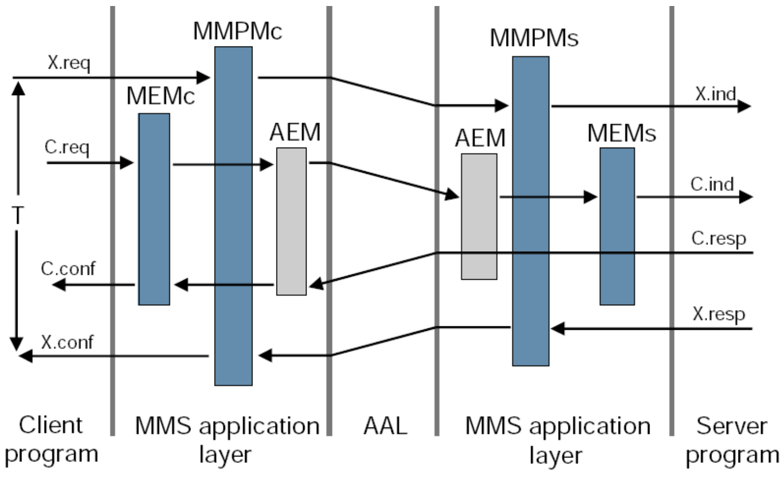

Fig. (6). Transmission of an MMS request and its response.

b) Measurements

Measurements have been classified into two categories:

1. *External measurements:* these consider the implementation as a closed box and provide response times for MMS request-responses between two MMS/ATM nodes.

2. *Internal measurements:* the implementation is evaluated from the inside to identify the time spent in different parts of the software. This includes PDU encoding/decoding, the MMPM state machine, ATM transfer and the VMD program.

i.) External Measurements

In this subsection, we focus on the performance perceived by user applications communicating via MMS. The measurements were evaluated by an MMS application program. The time for each service given in Table (1) was determined by sending a large number of them, recording the time for the total transmission and obtaining the average time per service request. The variables used in the server for experiments with the *Read* and *Write* services are of integer type with 2 bytes for the name and 4 bytes for value. Table 1 gives the request-response times of the Read, Write, Initiate, Conclude, Start, Stop, and Download services.

We note that as the number of variables read or written increases, the response time increases. This cannot be explained by the existence of the AAL or ATM layers. The reason for this increase is that the response time depends on the application layer and particularly on the encoding/decoding process. We have also measured the response time of the Download MMS service for two segment sizes: 512, 1024 bytes. Figure (8) indicates the Download time as a function of the segment number of 512 and 1024 bytes.

Table (1). Average Times for some MMS services.

MMS Service	**Time(ms)**
Read	4.30
Write	4.80
Initiate	4.70
Conclude	4.00
Start	4.30
Stop	4.10
Download	10.10

ii.) Internal Measurements

In the above paragraphs, we have considered the implementation of MMS-ATM as a black box that provides response time for MMS request-responses. This response time is determined by the execution time of five main components: MMPM, Encoder, Decoder, API and ATM transfer (Table 2).

Table (2). Execution Time for Read, Write and Download services.

Services	**READ**		**WRITE**		**DOWNLOAD**	
Component	TIME (MS)	*Percent*	TIME (MS)	*Percent*	TIME (MS)	*Percent*
VMD	0.50	11.6	0.60	12.5	0.60	5.9
MMPM	1.60	37.2	1.60	33.5	4.00	39.6
ENCODER	0.80	18.6	1.00	21.0	2.00	19.8
DECODER	0.60	14	0.80	16.5	1.50	14.9
ATM TRANSFER	0.80	18.6	0.80	16.5	2.00	19.8
TOTAL	4.30	100%	4.80	100%	10.10	100%

Figure(7) shows the execution time of Read and Write services of one variable (integer that has 2 bytes as name length and 4 bytes as value length).

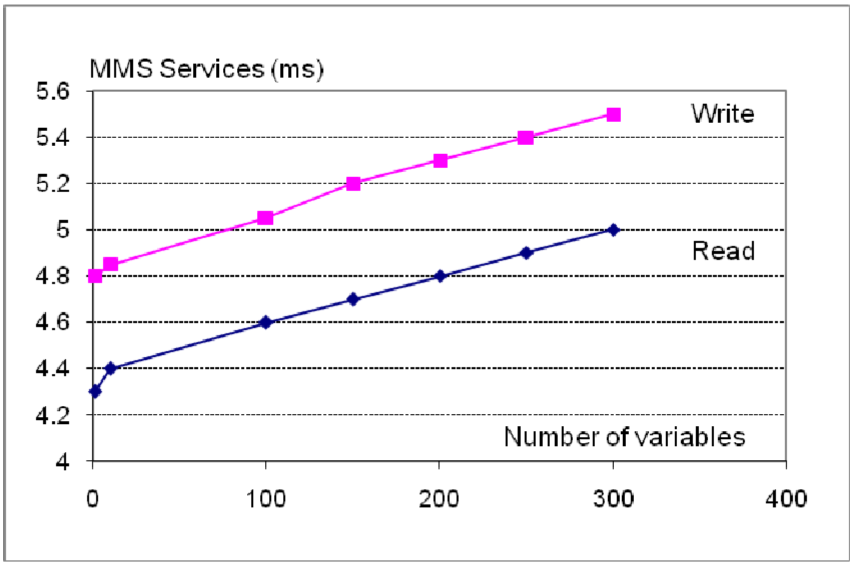

Fig. (7). Average Times for Read and Write services.

We have also measured the execution time of the MMS Domain service, which allows the transfer of data presented as segments. The request-responses time of the domain download service depends on the segment size. For a segment size of 512 bytes, the response time is 10.1 ms; detailed values of these measurements are given in fig.(8).

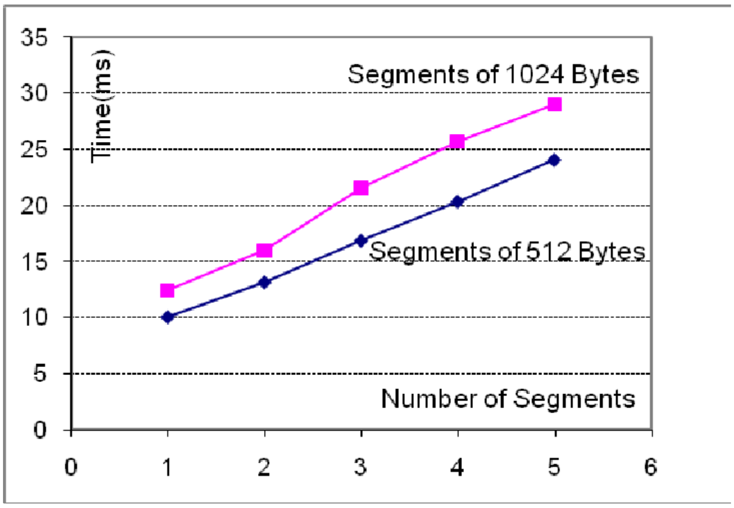

Fig. (8). Average Times for Download service.

5. Analysis and Comparison

a) Analysis

The external measurements show that the latency for the Download service is nearly twice as large as for others services. This cannot be explained by only the difference between the encoding/decoding time of *Download* service and the others services. The explanation is that the *Download* service involves twelve transfers through all the communication layers at the source and destination station. However for the other services there are only four transfers through all communication layers at the source and destination station.

Our implementation allows the execution of the Download service with segment sizes greater than 1024 bytes. For these segment sizes the response time increases significantly. This is the result of the number of transfers performed between the *emission/reception* buffers and the reassembly/segmentation buffers of our ATM communication system. For example, a 512 bytes segment requires two transfers and a 1024 bytes segment requires three transfers.

The internal measurements show that the MMPM execution time is greater than the encode/decode time together and is twice long as the ATM transfer time. This can be explained by the fact that MMPM connects seamlessly the MMS and AAL layers. MMPM runs some functions necessary for this connection such as: encapsulation, address mapping, and transfer. Details on the address mapping and encapsulation functions are provided in sections 3.3 and 3.4 respectively. The transfer function includes all low level transfer routines from the host to ATM-system.

b) Comparison between MMS/ATM and other architectures

Now, we try to compare the performance of our MMS/ATM communication system with the Mini-MAP and MAP architectures.

The Mini-MAP architecture considered consists of three layers: MMS application, data-link, and physical. The data-link layer is subdivided into two layers: the MAC (Medium Access Control) and the LLC (Logical Link Control) sub-layers. The MAC sublayer conforms to the IEEE 802.4 token passing bus standard (physical layer). The LLC sublayer offers two services: an unacknowledged connectionless service (LLC type 1) and an acknowledged connectionless service (LLC type 3). The Mini-MAP application layer offers the services of a reduced subset of OSI application service elements, namely MMS, network management and a directory called the object dictionary.

A major issue in manufacturing systems is the need for rapid and reliable communication among the wide variety of computers and automated equipment involved in the manufacturing process. Typically, the factory floor devices are incompatible with one another. A standardized communication network is needed to allow their interconnection.

MAP relies essentially on a standard connection oriented ISO profile. It implements all of the ISO protocol layers. Note that Mini-MAP MMS does not differ from MAP MMS in the syntax of messages but only in the underlying support called MMPM that governs the message exchanges between the *MAEs* [17].

We have compared our MMS/ATM implementation with some existing MMS implementations. In most cases our implementation has achieved a better or equivalent performance and it is advantageous [19]. This is due to:

- The use of ATM as the technology of a communication network has some technical advantages. In classical MMS/MAP or MMS/Mini-MAP implementations, the number of interconnected workstations is bounded, as well as that of logical segments, for example to 32 and 1 km respectively. In general, to interconnect several machines from different segments the designer may have to use bridges or gateways. This increases the network equipment and management cost. In contrast there are no such constraints in our MMS/ATM implementation.

Furthermore, it is worth to use ATM, since its virtual connection concept introduces flexibility in the management of the network connections.

- ATM networking provides end-to-end quality of service (QoS) guarantees [20]. It offers:
  - ✓ Transparent handling of different data types,
  - ✓ High requirements for multimedia applications (latency, bandwidth),
  - ✓ Multicast support
  - ✓ Real-Time support.
- Some modern applications of multimedia communication derived from the auto industry are used to describe the use of this technology in design, manufacturing and sales [3, 21]. Our MMS-ATM architecture offers high bandwidth (155 up to 622 Mbps) that allows integration of innovative applications as multimedia in industrial environments.

Some examples of multimedia applications in manufacturing systems are: *Tele-surveillance* and the *journal management*. The *Tele-surveillance* application in a manufacturing environment provides a means for the control of remote sites. Remote cameras and microphones are used to capture image and audio data that are sent to displays, loudspeakers, storage systems, or to specific signal processing systems. A Tele-surveillance application defined here is similar to a *robotics* application. The journal management application provides means for recording time-stamped image and audio information related to specific events that are likely to occur on a manufacturing plant.

These applications can be used to measure the positions of parts and guide the assembly process to reduce the need for costly and inflexible featuring. It can also be used to monitor and maintain quality of services parameters.

6. Possible Improvements

a) Introduction

The use of both software and hardware implementation in communication system seems to be a key issue in high performance networking [22]. Some parts of a communication system are implemented in hardware to achieve better performance, while other parts are implemented in software yielding more flexibility.

The goal of hardware/software design is to produce an efficient implementation that satisfies the performance and minimizes the cost, starting from the initial specification. However, it exists a diversity of technological solutions based on available hardware/software components. The designer needs to explore the possible solutions either using automated tools or selecting his choices manually. Different solutions are analyzed to choose the best one from trade-offs between both technologies. The latter result is the one that satisfies best the required performance. At any stage of the design, the user can cancel one or several design steps in order to explore new choices.

The MMS/ATM communication system presented in this paper is composed of two parts: hardware and software. We have evaluated and analyzed in section 4.2 the performance of such a system. As a result, we have noted that the transmission delay of MMS services depends on the performance of the software part. Our objective is to improve the transmission delay of MMS services by implementing in hardware/software solution. This software part contains some critical tasks such as encoding and decoding process.

b) Transmission delay

In this subsection, we study the transmission delay of MMS services. We are interested in the delay caused by the execution of protocol layers, in particular the execution of MMS/ATM communication sublayer. We calculate the percentage of the execution time in this sublayer with respect to the total time of the MMS service transmission delay. This percentage is calculated for different robot response times.

The transmission delay of the MMS *Write* service is composed of seven laps of times (Fig.1). t1 and t6 are the execution times of client and server application entities, respectively. t3 and t4 represent the transmission times between two MMS/ATM nodes. The execution time of the MMS/ATM communication sublayer at client and server node is, respectively, t2 and t5. Finally, t7 is the response time of industrial equipment (e.g. Robot).

We have defined three functions te, tc and R:

- ✓ Execution time of MMS/ATM communication software part te = 2*(t2 + t5) (MMS/ATM communication sublayer),
- ✓ Execution time of the MMS Write service tc = te + 2*(t1 + t3 + t4 + t6) + t7,
- ✓ The ratio: R = te/tc

When the response time of a robot is 100 ms, the execution time (te) of the MMS/ATM communication sublayer which is a software part is not significant. But this execution time (te) becomes significant (*39%<R<60%)* when the response time of a robot is less than 10 ms. In this case, a hardware/software implementation of the MMS/ATM communication sublayer is recommended to satisfy the real-time requirements of industrial applications.

By implementing an MMS/ATM rather than an MMS/TCP/IP/ATM, we thought, we have implemented a faster communication system. However, Fig.(9) indicates that the overhead incurred due the MMS/ATM communication sublayer, which is a software implementation, can reach 60% of the overall communication time. This justifies a hardware/software implementation of this sublayer, hoping to achieve better performance.

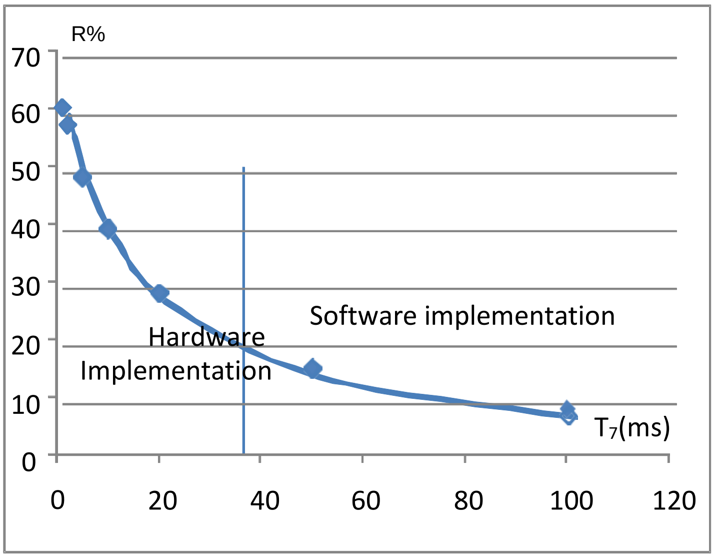

Fig.(9). Write execution time Ratio with different t_7 values.

7. Conclusions

In this paper we have studied the implementation structure and the resulting performance of an MMS/ATM communication system. Then we evaluated the performance of this implementation from an external and an internal point of view.

The external measurements show the response time of MMS services which is high for some application's critical-time. We have expected that the use of ATM technology (high speed) and the minimization of the number of protocol layers will yield small response times. To understand the partition of this transmission delay in each part of our MMS/ATM communication system and to propose a solution to decrease this delay, we have carried out internal measurements.

The internal measurements have shown that most of the MMS request-response time (80%) is spent in the MMS/ATM application layer stack. Then, we have studied a hardware/software implementation of the MMS protocol. This kind of implementation allows improving optimal cost of the transmission delay of MMS services. As a result, the time spend in the application layer becomes 60% per cent of the MMS request-response time.

Experience made has shown that memory management is a key factor for the performance of the encoding and decoding process. We have chosen to implement the encoding/decoding process in hardware on a co-processor. This hardware implementation allows a better encoding/decoding performance avoiding any memory management problems.

8. References

[1] Jan Tore Sørensen, Martin Gilje Jaatun, **"**An Analysis of the Manufacturing Messaging Specification Protocol**",** *Proceedings of the 5th international conference on Ubiquitous Intelligence and Computing*, Oslo, Norway, (2008),pp 602-615.

[2] ISO/IEC. 9506-1, *Industrial Automation Systems-Manufacturing Message Specification, Part 1: Protocol Definition*, International Standards Organization, (2003).

[3] ISO 9506-2, *Industrial Automation Systems-Manufacturing Message Specification*, *Part 2: Service definition*, International Standards Organization, (2003).

[4] Suzuki, K.; Tetsuo Ideguchi, "Proposal of Device Capability Profiling for Manufacturing Application Service Interface", *IEEE International Conference on Industrial Technology*, Monash University, Victoria, Australia, 10-13 Feb. (2009), pp 1-6.

[5] IEC-TC 57, *Communication Networks and Systems in Substations -Part 8-1: Specific Communication Service Mapping (SCSM) - Mapping to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3,* Edition 1.0, IEC Standard IEC/TR 61850-1, (2004).

[6] Jain, M.; Vidyasagar, L., Rajashekhar, N.H., Ramamohan, T.R.; Pranesh, T.N, "ATM Protocol Adaptation for EMI/EMC environment", *Proceedings of the International Conference on Electromagnetic Interference and Compatibility*, 21-23 Feb. (2002), Bangalore, India, pp 187-192.
[7] Bhar, J.; Ouni, R.; Helali, A.; Nasri, S.; "Improvements of the ABR loop performances in a wireless ATM network" *Proceedings of the 16th International Conference on Microelectronics*, ICM 2004. 6-8 Dec. 2004, Page(s):295 – 298, Tunis, Tunisia.
[8] Wu Renyong; Duan Zhenghua; "ATM protocol on LAN" *IEEE International Conference on Intelligent Processing Systems, ICIPS '97.*, 28-31 Oct. (1997), Vol.1, Beijing, China, pp 949 – 952.
[9] R. Jain, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey", *Computer Networks and ISDN Systems Journal*, Vol. 28, No 13, (1995), pp 1723-1738
[10] Qin Zheng; Shin, K.G.; Chia Shen, "Real-time communication in ATM networks", *Proceedings of 19th Conference on Local Computer Networks*, 2-5/10/1994, Minneapolis, MN, USA, pp 156 –164,
[11] K. Schwarz, *Introduction to the Manufacturing Message Specification (MMS, ISO/IEC 9506)*, Web Document, Netted Automation, 2000, viewed June (2005).
http://www.nettedautomation.com/standardization/ISO/TC184/SC
5/WG2/mms_intro/
[12] R. Mackiewicz, *An Overview to the Manufacturing Message Specification*, Engineering Research Center for Advanced Control & Instrumentation, Seoul National University, Korea, (1994), Reviewed June (2005), http://erc-aci
.snu.ac.kr/mms/mms_IntroSISCO.html
[13] Martin de Prycker, *Asynchronous Transfer Mode*, Ellis Horwood, Upper Saddle River, NJ, USA, ISBN:0-13-053513-3, (1991)
[14] Sumit Kaser, *Atm Networks Concepts And Protocols*, McGraw-Hill, Tata, 2006, ISBN : 0070583536, 9780070583535
[15] ITU-T, *Recommendation I.363 - B-ISDN ATM Adaptation Layer (AAL) Specification*, March (1993).
[16] A. Valenzano and L. Ciminiera, "Performance Evaluation of Mini-MAP Networks", *IEEE Transaction on Industrial Electronics*, Vol. 37 (3), (1990), pp 253-258.
[17] M. Li, P. Pleinevaux and F. Vamparys, "Performance Evaluation of a Reduced OSI Stack", *Internetworking: Research and Experience*, Vol. 5, pp 71-87. (1994)
[18] ISO 8825, *Information Processing Systems, Open Systems Interconnection, ASN.1 Basic Encoding Rules*.
[19] P. Castori, P. Pleinveaux, "A Generic Architecture for MMS Servers", *Proceedings of the IEEE International Workshop on Factory Communication Systems (WFCS)*, Leysin, Switzerland, (1995), pp 211-218.

[20] C. Shen, “On ATM Support for Distributed Real-Time Applications”, *Proceedings of the 2nd IEEE Real-Time Technology and Applications Symposium, (RTAS '96)*, June 10 - 12, (1996), Boston, MA, USA.

[21] Chwan-Hwa "JOHN" WU, M. C. YUANG and J. D. IRWIN, “Multimedia and Multimedia Communications: A Tutorial”, *IEEE Transaction on Industrial Electronics*, Vol. 45 (1). (1998).

[22] S. FISCHER, J. WYTREBOWICZ, S. BUDKOWSKI, “Hardware/Software Co-design of Communication Protocols”, *Proceedings of EUROMICRO’96*, September 2-5, (1996), Prague, Czech Republic.