ELSEVIER
IFAC
PUBLICATIONS
www.elsevier.com/locate/ifac

A CANOPEN-BASED PROTOCOL FOR DYNAMIC ASSIGNMENT OF PRIORITIES

Salvatore Cavalieri

University of Catania, Faculty of Engineering
Department of Computer and Telecommunications Engineering
Viale A.Doria, 6 95125
Catania (Italy)
E-mail: Salvatore.Cavalieri@diit.unict.it

Abstract: The author deals with the problem of information flow management in a CANOpen network. CAN ISO IS-11898 physical medium access mechanism, adopted in CANOpen, features an arbitration protocol based on an identifier assigned to each message to be transmitted. The lack of a centralised scheduling for the identifier assignments in CANOpen does not allow achieving any guarantee to meet the deadlines of time-critical messages. The aim of the paper is to propose a protocol for dynamic and centralised assignment of identifiers to each message to be transmitted, in such a way that its transmission requirements are fulfilled. The main features of the protocol proposed are the operational flexibility it offers and full compatibility with the CANOpen standard. *Copyright © 2003 IFAC*

Keywords: CANOpen, Fieldbus, Identifiers, Scheduling, CSMA/CA

1. INTRODUCTION

The main feature of the CAN fieldbus is the Carrier Sensor Multiple Access/Collision Avoidance (CSMA/CA) mechanism based on the presence of a dominant bit (zero) (ISO IS-11898, 1993). This mechanism is applied to an identifier field univocally associated to each message to be transmitted.

CAN is actually used in wide range of industrial applications. Most of them are classified as Real-Time applications whose correctness depends not only on the logical results of computing, but also on the time at which the results are produced (Arvind, Ramamritham and Stankovic, 1991).

In Real-Time applications, messages must be properly scheduled to access to the network, in order to have their deadlines respected. Literature presents several scheduling algorithms in CAN. Many of them are based on the idea to link the message identifier with its transmission time features and requirements. One of the most known approach sets the identifier to a unique priority, according to the Deadline Monotonic rule (Tindell, Hansson and Wellings, 1995) (Tindell, Hansson and Wellings, 1994). The reader is referred to (Tovar, 1999)(Ouni and Kamoun, 1999) for other examples of scheduling approaches in CAN.

When the identifiers of messages must be assigned according to a particular rule (e.g. deadline monotonic, as said before), the need for a protocol able to assign and to on-line manage the message identifiers (coping with reconfiguration, addition and deletion of nodes, addition and deletion of messages produced by a node already existing) arises. The definition of such protocol is much more difficult in a CAN network made up by a standard Application Layer above the ISO IS-11898 protocol, because the protocol for the dynamic assignment of identifiers must be compliant with the services available at that layer.

In the previous work (Cavalieri, 2001), the author proposed a protocol for the dynamic assignment of identifiers fully compliant with the CAL CiA standard (CiA/DS201, 1996). The aim of the paper is

to extend this work to the CANOpen communication system (CiA/DS301, 1996)(Cenelec EN50325-4, 2001). Although it derives from CAL CiA standard, CANOpen offers different services and may not provide for some features present in CAL CiA. For example, in this last standard a particular entity, called DBT Master (CiA/DS204-1, 1996)(CiA/DS204-2, 1996), is responsible to distribute the identifiers to each message in the network. The protocol proposed in (Cavalieri, 2001) is fully based on this entity. The not mandatory presence of DBT Master in CANOpen requires the re-definition of an identifier assignment policy for the CANOpen system.

On the basis of what said, the aim of this paper is to propose a protocol able to cope with on-line management of identifiers in CANOpen communication system. The protocol is able to assign identifiers to new nodes, added to the network, and to reconfigure identifiers when necessary. Further, it is full compliant with the CANOpen standard (CiA/DS301, 1996)(Cenelec EN 50325-4, 2001). No modifications to the CANOpen are required, as the protocol proposed is just an add-on.

The paper will clearly describe the dynamic identifier assignment protocol and will demonstrate its compatibility with the CANOpen standard. Finally a performance evaluation will be presented in order to point out the capability of the protocol to assign the identifiers and to respect all the real-time constraints of the messages to be transmitted.

2. A BRIEF OVERVIEW OF THE PROTOCOL

The protocol here proposed is based on the definition of a centralised policy for the assignment of identifiers to messages in the CANOpen communication system. In this standard, the identifies are called Communication Object Identifiers (COB-ID). It is assumed that one of the communication nodes in a CANOpen acts as master and manages the assignment of the COB-IDs. The node responsible for this is called the Schedule Master (SM) and the other nodes are the Schedule Slaves (SS). They are the communication nodes connected to sensors/actuators/controllers, so producing information flow featuring both real-time (RT) and non real-time (NRT) constraints. As known, Real-time traffic may be further divided into hard real-time (HRT) and soft real-time (SRT).

The SM receives information from each SS about the traffic it generates (e.g. real-time features such as period and deadline). The SM executes a suitable scheduling algorithm able to assign COB-IDs to each SS in such a way time requirements of each SS are satisfied through the contention mechanism featured by the CAN Data Link Layer. Finally the SM will distribute the assigned COB-Id to each SS. If one or more time requirements cannot be met by COB-ID assignment, the SM will activate a procedure to reassign all or a part of COB-IDs, in order to meet the unfulfilled requests.

It's important to point out that no scheduling algorithm is here proposed, as it goes beyond the aim of the paper. Section 4.3 will give some details about the features that a scheduling algorithm must offer.

The proposed protocol features different behaviours according to the operational phase of the CANOpen communication system: at system start-up and run-time. On system start-up, when no nodes have as yet obtained COB-IDs, the SM collects the list of the time constraints to be fulfilled from each node connected to the network. Then the SM runs a scheduling algorithm in order to assign the COB-ID to each request and distributes the COB-IDs to the SSs. At run-time, when a new node is added to the network and/or an SS that is already active needs to obtain new COB-IDs for processes that have been added to it, the SM has to collect the COB-ID assignment requests from the new node or the existing one and has to assign and distribute the COB-IDs to it. At system run-time a re-assignment procedure must be also played by the SM, when one or more COB-ID don't fulfil time requirements of the relevant time critical processes.

3. COMMUNICATION IN CANOPEN STANDARD

According to (Cenelec EN 50325-4, 2001) and (CiA/DS301, 1996) communication in CANOpen is realised through two types of communication objects: Process Data Object (PDO) and Service Data Object (SDO).

The real-time data transfer is performed by means of PDO on a producer/consumer(s) basis. There are two kinds of use for PDOs. The first is data transmission (realised through Transmit-PDO) and the second data reception (through Receive-PDO). Devices supporting Transmit-PDO are PDO producers and devices which support the Receive-PDOs are called PDO consumers. Each PDO (Transmit and Receive) corresponds to entries in the device Object Dictionary hold by each device in a CANOpen network. The Object Dictionary is essentially a grouping of objects accessible via the network in an order pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. Each PDO is fully described by a Communication Parameter and a Mapping Parameter. Communication Parameter specifies, among others, the COB-ID used for the exchange of information. Mapping parameter specifies the objects mapped to the PDO-based information flow. Communication and Mapping Parameters corresponds to entries in the device Object Dictionary. In CANOpen for each device it's possible to configure at most 512 Transmit and 512 Receive PDOs. PDO transmission shall follow the producer/consumer relationship. One of the services allowing a producer to send data to consumer(s) is the WritePDO one, that will be used in the proposal here presented.

SDO is used to transfer larger data set, containing arbitrary large block of data. Another difference with PDO is that SDO communication is based on a client/server model. Basically an SDO is transferred as a sequence of segments. Like PDO, each SDO corresponds to entries in the device Object Dictionary. Two kinds of SDO entry are present in the Object Dictionary: the Server and the Client SDO Parameter. The last is used by the requesting node, while the first by the responding node. In CANOpen, for each device it is possible to configure at most 128 Server PDO Parameters and 128 Client SDO Parameters. Among other things, a Server/Client SDO Parameter specifies the COB-IDs used for the exchange of information from client to server and from server to client. One of the services used in the SDO-based communication is the SDODownload, that will be used in the proposal.
Like all CAN-based communication systems, all the messages sent in the network features a priority given by the COB-ID. Considering a 11-bit ID CAN network, the COB-IDs available for the communication are 1760, ranging from 1 to 1760.
The COB-ID assignment in CANOpen network may be pre-configured and/or dynamic. The adoption of the services defined in CAL CiA (CiA/DS201, 1999)(CiA/DS204-1, 1996)(CiA/DS204-2, 1996) to dynamically manage the COB-ID assignment is foreseen. In this case, a particular entity called DBT Master plays the role to assign and distribute identifiers to each communication node, maintaining these identifiers into a particular database. As can be found in (CiA/DS301, 1996), DBT Management is not mandatory in CANOpen communication system. For this reason, it wasn't considered in the proposal here presented.

4. A DETAILED DESCRIPTION OF THE PROTOCOL PROPOSED

The aim of this section is to present a more detailed description of the protocol proposed for COB-ID assignment. It will be divided into three parts, referring to the actions required for the assignment of new COB-IDs at system start-up, the assignment of COB-IDs to new nodes or to new processes added to already existing nodes, and to the COB-ID re-assignment procedure.

4.1 COB-ID Assignment at System Start-up

At the system start-up, according to the protocol here proposed, the SM is the only node in the network enabled to start the communication; all the other nodes (i.e. SSs) are in a stand-by condition, listening for any command sent by the SM and reacting to this command as explained in the following.
The SM polls all the nodes, sending to each one of them a particular message containing the information about the notification that the sending node is the SM, its address (i.e. node ID) and the command to receive the COB-ID assignment requests. After the SM has sent this message to a particular node, it waits until a time-out expires. If the SM doesn't receive any request from the SS, it will pass to poll the next node in the network. The need for a time-out for the SM is clear as the CANOpen communication systems features the presence of 127 nodes, but a lower number of nodes may be present at system-start-up.
The service used by the SM to perform this protocol is the WritePDO service provided for by the Process Data Object (PDO) (CiA/DS301, 1996). With this Store-Events type service the SM will compel each SS to transmit COB-ID assignment request(s).
If the SS is connected to the network reacts to the previous service by preparing and sending one or more COB-ID assignment requests, using SDODownload service provided for the Service Data Object (SDO).
When the SM will collect all the requests coming from the SSs connected to the network, it will run a suitable scheduling algorithm, in order to evaluate the COB-IDs to be assigned to each SS, able to respect the relevant time requirements and constraints. Finally, the SM will distribute these COB-IDs to each SS using again the SDODownload service.
Figure 1 shows the sequence of messages generated on the network. As can be seen, the figure shows also the COB-IDs used to transmit the services WritePDO and SDODownload. As COB-ID assignment is not featured by real-time requirements, it was assumed to use COB-IDs whose values belong to an intermediate priority class, considering that COB-ID range is from 1 to 1760.

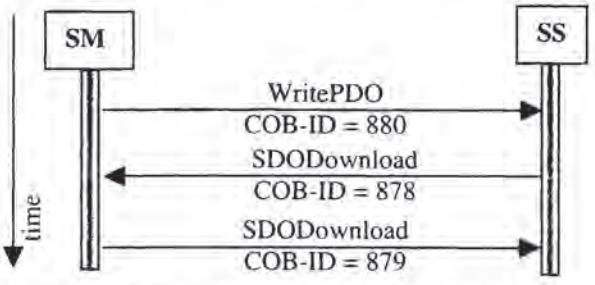

Figure 1 - Sequence of messages involved in the request and assignment of COB-IDs at system start-up.

4.2 Run-Time COB-ID Assignment

After the system start-up, if a new node is added to the network, it can't start any activities. It has to wait for an explicit command from the SM of the network. For this reason, the protocol here proposed foresees that at run-time the SM periodically polls all the nodes whose node ID is not in its list of active nodes. This list has been built at system start-up during the polling procedure described in the previous sub-section.

When the polling involves a new added node, this last can send all its time requirements and can receive the relevant COB-IDs from the SM. The exchange of information between SM and the new added node is quite similar to that depicted in Figure 1. In this case the WritePDO service is used by the SM to poll the nodes not included in the list of active nodes. The SDODownload services are used to collect COB-ID assignment requests from SSs and to distribute the COB-IDs to the SSs.

Let's now consider another scenario, featured by the presence of a SS already present in the network (i.e. whose node ID is included in the list of active nodes hold by the SM), requesting new a COB-ID assignment. This happens when one or more processes are started within the SS. The SS will notify this event to the SM by a WritePDO service. With this Store-Events type service, the SM will be aware of the COB-ID assignment need featured by the particular SS. After this notification, the SS prepares and sends an explicit COB-ID assignment request, using the SDODownload service.

When the SM will receive the requests coming from the SS, it will evaluate the COB-IDs and will distribute them to the requesting SS by the SDODownload service.

Figure 2 shows how the COB-ID request and assignment protocol at run-time is implemented in the case an already existing node requires other COB-IDs to be assigned, pointing out the COB-IDs used for the information flow.

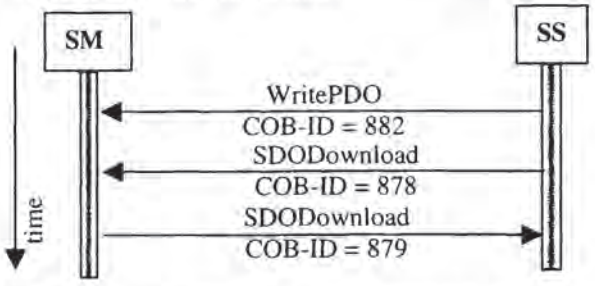

Figure 2 - Sequence of messages involved in the run-time request and assignment of COB-IDs.

4.3 Run-Time COB-ID Re-Assignment

COB-ID re-assignment procedure is needed when one or more "unsatisfactory" COB-ID assignments happen. Definition of unsatisfactory assignments can't be given if general criteria about the COB-ID assignment algorithm are not pointed out.

As said before, the definition of the scheduling algorithm played by SM for the COB-ID assignment is beyond the aim of the paper. Literature presents a lot of scheduling algorithms, and the reader can easily find the most suitable to be used in the proposal here presented. In order to do this, the main features the scheduling algorithm has to hold, must be clearly defined.

First of all, the algorithm must base the scheduling procedure on the time requirements of the requesting process (i.e. the hard, soft and non-real time kind of information flow, its period and deadline, if any). Then, the COB-IDs must be assigned in such a way the highest priority COB-IDs (i.e. the COB-IDs with the lowest values) are assigned to the hardest real time processes with the shortest deadline.

It's clear that there is the possibility that a COB-ID value corresponding to a particular kind of traffic and a particular deadline is not available during COB-ID assignment procedure. For example let assume that COB-ID values belonging to the range [900-1000] have been already assigned to hard-real time process with deadline ranging from 7 to 10 msec. Further, let assume that no free COB-ID are available in this range. If a new hard real time process featured by a deadline of 8 msecs requests a new COB-ID assignment, a COB-ID outside the range of [900-1000] must be assigned.

In this case, the SM has two possibilities: a higher priority COB-ID or a lower priority COB-ID could be assigned to the requesting SS. There are no constraints on the choice performed by SM; choosing a higher priority COB-ID the SM will satisfy the time constraints of the requesting node, but it will introduce the possibility that more critical processes can't find free COB-IDs in the future. On the contrary, choosing a lower priority COB-ID means that the time constraints of the requesting SS may be not satisfied. In some cases, the SM is obliged to choose a lower priority COB-ID, for example when no higher priority COB-IDs are available at the moment. If the SM chooses to assign a COB-ID featured by a lower priority than that relevant to the assignment request then an "unsatisfactory" COB-ID assignment arises.

Coming back to the previous example, and assuming that the SM has assigned to the hard real time process featured by a deadline of 8 msecs a COB-ID higher than 1000, generally it's not possible to state that the deadline of the process will be certainly missed. Several techniques to ascertain this are available in literature. For example in (Tovar, 1999) a worst-case response time evaluation technique has been shown. It's clear that whatever procedure for investigating the respect of the deadline for each "unsatisfactory" assignment cannot be realised on-line during COB-ID assignment procedure.

So in the proposal, it was foreseen that periodically the SM evaluates all the previous "unsatisfactory" assignments. If a COB-ID not able to respect the deadline of the process is found, then a reassignment procedure is activated by the SM. The actions taken by the Schedule Master are as follows:

1 reassigning the "unsatisfactory" COB-ID to a higher priority value (if COB-ID values are now available), evaluating the respect of the deadlines;
2 sending to the SS relevant the reassigned COB-ID a new COB-ID, using the WritePDO service, as depicted in Figure 3. The figure shows the COB-ID used in the information flow.

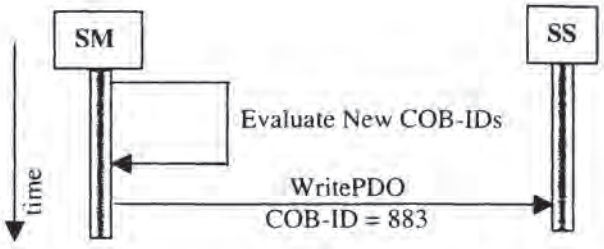

Figure 3 - Operations and messages on the network following a re-assignment event.

5. A DETAILED DESCRIPTION OF THE SERVICES REQUIRED TO IMPLEMENT THE PROPOSAL

The aim of this section is to give details of the services and parameters used to implement the dynamic identifier assignment protocol described in the previous section. As said, both PDO and SDO-based services available in (Cenelec EN 50325-4, 2001) have been used. According to the standard, their use implies the need to define particular entries in the Object Dictionary of the devices (SSs and SM) and to reserve COB-IDs for the information flow needed to implement the protocol here proposed.
The description of the PDO and SDO-based services used in the COB-ID assignment protocol will be made according to the various phases of the protocol.

5.1 System Start-up

As said in Section 4.1, the COB-ID assignment at system start-up has been realised using two services: WritePDO and SDODownload. Further, it was said that the exchange of the service WritePDO was realised using COB-ID 880, and exchanges of service SDODownload using COB-IDs 878 and 879.
WritePDO is used by the SM to notify to each SS its presence and to request COB-ID assignment requests from SS. WritePDO needs two arguments:

- the PDO Number of the TransmitPDO parameters in the Object Dictionary of the sending SM and of the ReceivePDO parameters in the Object Dictionary of each receiving SS;
- the data to be transmitted.

In the proposal the following arguments have been used:

WritePDO(1, event_collect)

The first argument, as said before, refers to the PDO Number of the Transmit/Receive-PDO. In the paper it was assumed to use the PDO Number 1 to realise the communication between SM and SS at system start-up. In particular, the SM has the Transmit PDO number 1 stored in its OD. It's featured by a PDO Communication parameter at the index 1800h and a PDO Mapping Parameter at index 1A00h. In the PDO Communication parameter the COB-ID number 880 is associated. Each SS has the Receive PDO number 1 stored in the OD. It's PDO Communication parameter is at index 1400h and contains the COB-ID 880.
The second argument, event_collect, represents the data to be transferred. This data is a record made up by the fields: Node-ID of the sending SM and the Node-ID of the SS to which the service is sent.
SDODownload is used by each SS to send to the SM the COB-ID assignment requests. In this case it was assumed to reserve the COB-ID 878 for the exchange of this service. SDODownload is featured by several arguments, among which the most important to be defined are:

- the SDO Number of the ClientSDO parameter for the request from a client, and the SDO Number of the Server SDO parameter for the response from the Server; in the proposal it was assumed that the SM plays the role of Server and each SS that of Client;
- the data to be transmitted.

In the proposal the following arguments have been used:

SDODownload(1, schedule_data)

The first parameter refers to the ClientSDO Number stored in both SM and each SS. According to CANOpen (Cenelec EN 50325-4, 2001) the first ClientSDO Number corresponds to an entry at index 1280h in the Object Dictionary of each device.
The second parameter, schedule_data, represents the data to be transferred. It is a structure featuring the following fields:

- RTtype indicates the type of real-time service requested by the SS: HRT, SRT, or NRT;
- Deadline indicates the deadline of the periodic process (valid only for HRT and SRT services, ignored for NRT services);
- Period indicates the period of the periodic process. A value of 0 is used to indicate asynchronous processes;
- NodeID identifies the number of the module and can take a value between 1 and 127;
- ProcessID identifies the process on the node and can take a value between 0 and 65535;
- COBname is a string of 14 characters containing the name of the COB.

SDODownload service is also used in the proposal for the distribution of the COB-IDs to each SS by the SM, after it has collected all the COB-ID assignment requests from SS and has performed the relevant assignment. In this case it was assumed that the exchange of information from SM to SS is realised using the reserved COB-ID 879. Further the following arguments of the SDODownload service have been used:

SDODownload(2, schedule_result)

The first parameter refers to the ServerSDO Number stored in both SM and each SS. The second ServerSDO Number corresponds to an entry at index 1201h in the Object Dictionary of each device.
The second parameter, schedule_result, represents the data to be transferred from the SM to each SS. It is a structure featuring the following fields:

- NodeID identifies the number of the module to which the reply is addressed;
- ProcessID identifies the process on the node that has requested the COB-ID;
- COBname is the name of the COB;
- COBID is the identifier assigned by the SM.

5.2 *Run-Time*

As explained in Section 4.2, a new node added into the CANOpen network or an already existing SS, may advance COB-ID assignment requests. The protocol here proposed features the WritePDO service used by the SS to notify its need to the SM, and the SDODownload service to collect the COB-ID assignment requests from each SS and for the COB-ID distribution to each SS by the SM. Exchange of WritePDO service from SS to SM is realised using the reserved COB-ID 882, while the exchange of SDODownload is performed using COB-IDs 878 (from SS to SM) and 879 (from SM to each SS).
In the proposal the following arguments have been used for the WritePDO service exchanged from SS to SM:

WritePDO(1, event_novel_assignment)

The first argument, as said before, refers to the PDO Number of the Transmit/Receive-PDO. In the paper it was assumed to use the PDO Number 1 to realise the communication between SS and SM at run-time. In particular, each SS, which now is the initiating node, has the Transmit PDO number 1 stored in its OD. It's featured by a PDO Communication parameter at the index 1800h and a PDO Mapping Parameter at index 1A00h. In the PDO Communication parameter the COB-ID number 882 is associated. The SM has the Receive PDO number 1 stored in the OD. It's PDO Communication parameter is at index 1400h and contains the COB-ID 882.
The second argument, event_novel_assignment, represents the data to be transferred. This data is a structure made up by only one field represented by the Node-ID of the sending SS.
About the arguments of the services SDODownload used to collect COB-ID assignment requests from the each SS and to distribute the new assigned COB-IDs to each SS, they are the same used in the system start-up phase of the protocol.

5.3 *COB-ID Re-assignment*

During COB-ID assignment procedure, the SM may realise that one or more COB-IDs are not compliant with the real-time constraints of the requesting node. In this case it was assumed that periodically, the SM try to adjust the COB-ID assignments, changing one or more COB-IDs in order to better respect time constraints of a particular node. The exchange of information between SM and SS to send the new assigned COB-ID is realised by the WritePDO service, as explained in Section 4.3. Exchange of WritePDO service from SS to SM is realised using the reserved COB-ID 883.
The following arguments have been used for the WritePDO service exchanged for the COB-ID reassignment purpose:

WritePDO(2, event_reassignt)

The first argument refers to the Transmit/ReceivePDO Number. The PDO Number 2 is used to realise the communication between SM and SS for the COB-ID reassignment. The SM, which is the initiating node, has the Transmit PDO number 2 stored in its OD. It's featured by a PDO Communication parameter at the index 1801h containing the COB-ID number 883, used for the information flow. Each SS has the Receive PDO number 2 stored in the OD. It's PDO Communication parameter is at index 1401h and contains the COB-ID 883.
The second argument, event_reassign, represents the data to be transferred. It is a structure featuring the following fields:

- NodeID identifies the SS to which the new COB-ID is sent;
- OLD_COB-ID is the COB-ID that must be changed;
- NEW_COB-ID is the reassigned COB-ID.

6. PERFORMANCE EVALUATION

The aim of this section is to present an assessment of the COB-ID assignment strategy here proposed. The assessment has been carried out taking into account that the main aim of the COB-ID assignment procedure has to be the respect of the time constraints (e.g. deadline) of the real-time messages. So the assessment was finalised to verify the capability of the proposal to respect all the real-time constraints of the requesting node in a CANOpen network.
As said in the Introduction, literature presents a lot of paper about scheduling solutions in CAN-based systems. However, these solutions are generally based on the CAN ISO IS-11898 protocol adopting a proprietary solution at the Application layer. On the other hand, the proposal here presented is based on a standard Application layer (i.e. the CANOpen). For this reason a comparison (in terms of capability to respect real-time constraints) between the scheduling solution proposed in this paper and those available in literature (based on CAN ISO IS-11898) hasn't considered. In fact, in the case of some kind of difference in the assessment, it would be very difficult to attribute this difference to the algorithm featuring the scheduling solution or to the number (and kind) of communication layers on which the scheduling solution is based.
Comparison with the optional COB-ID assignment strategy that could be used in CANOpen, have been assumed to be more meaningful in order to evaluate the scheduling solution here proposed. As can be

found in (CiA/DS301, 1996), CANOpen may foresee the presence of a COB Database, containing registration of all the COB-IDs transiting through the network. A particular entity called DBT Master is responsible to assign COB-ID and make the relevant registration into the COB Database (CiA/DS204-1, 1996) (CiA/DS204-2, 1996). As already said, use of DBT Management is not mandatory in CANOpen communication system. So it could be very interesting to compare the capability in respecting the real-time constraints of the information flow in a CANOpen based on the use of DBT Master and in a CANOpen where the COB-ID procedure here presented runs. In this case, comparison is meaningful as both solutions are based on the same Application layer.

A CANOpen communication system, the proposed COB-ID assignment strategy and the optional COB-ID assignment strategy based on DBT Management have been modelled using Estelle (ISO 9074, 1997). The use of Estelle has also allowed to validate the COB-ID assignment protocol presented in this paper.

Simulations have been carried out varying different parameters, like the number of variables to be transmitted and the relevant time constraints. For each simulation COB-ID assignment policies (that proposed in the paper and that offered by the CANOpen) have been evaluated verifying that all the time requirements taken into consideration have been fulfilled.

Performance evaluation and comparison between the proposal presented in the paper and the DBT management foreseen in the CANOpen has been performed according to the procedure described below.

Once a particular communication scenario made up by a bit rate value of the communication system, a certain number of variables to be transmitted, their time constraints (real-time kind, deadline, period) has been fixed, simulation is performed to obtain the COB-ID assignment realised according to the approach here presented and that offered by the standard CANOpen (using DBT services of CAL CiA).

On the basis of the COB-IDs assigned to each variable, both in the start-up phase and in run-time, the worst-case response time analysis presented in (Tovar, 1999) is applied. This analysis allows obtaining the worst-case response time value of each variable. This value has been compared to the relevant deadline, in order to check if this last has been respected.

In the following, some of the most significant results of the performance evaluation carried out will be presented.

Table 1 summarises the communication scenario taken into consideration. As can be seen, 600 variables have been considered, all featured by hard real-time constraints. Length, period and deadline of the variables are shown in the table. A CANOpen communication network running at 100Kbps has been considered. Then it was assumed that at system start-up the 200 variables featuring deadline of 220 msec and the 200 variables with deadline of 400 msecs were present. At run-time the other variables shown in the table were assumed to be added to the system.

Table 1 - Communication Scenario

Number of Variables	Period	Deadline	RT Type	Size bytes
200	220 msec	220 msec	hard	2
200	400 msec	400 msec	hard	2
100	125 msec	125 msec	hard	2
100	500 msec	500 msec	hard	2

Figure 4 shows the Worst Case Response Time (WCRT) versus the deadline of each variable (each value in the abscissa represents a variable, the first 400 of which are those present at system start-up, as said).

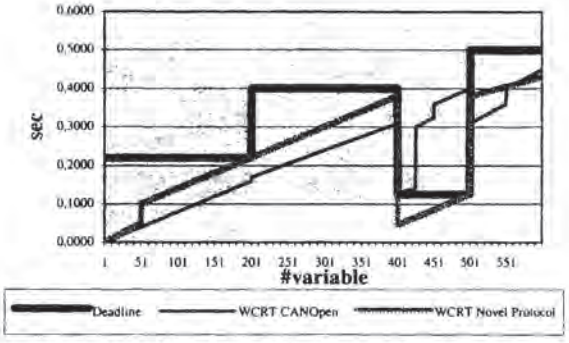

Figure 4 - Worst Case Response Time Evaluation

Worst case response time is depicted considering the proposal here presented (WCRT NovelProtocol) and the standard CANOpen (WCRT CANOpen). As can be seen, the protocol here proposed assures that all the time constraints are satisfied. Using DBT services of CAL CiA, some deadlines are not respected (in particular the 125 msecs deadline of the 100 variables added at runtime).

Another simulation was performed taking into account the communication scenario shown in Table 1. The only difference is that now it was assumed that at system start-up 150 variables featuring deadline of 220 msec and 150 variables with deadline of 400 msecs were present. At run-time, the other 50 variables featuring deadline of 220 msec, the other 50 variables with deadline of 400 msecs and the other variables shown in the table 1 were added to the system.

Figure 5 shows the comparison of the Worst Case Response Time considering the protocol here proposed (WCRT NovelProtocol) and that based on DBT services (WCRT CANOpen) versus the deadline of each variable. As can be seen, again the protocol here proposed assures that all the time constraints are satisfied. Using DBT services of CAL CIA, some deadlines are not respected.

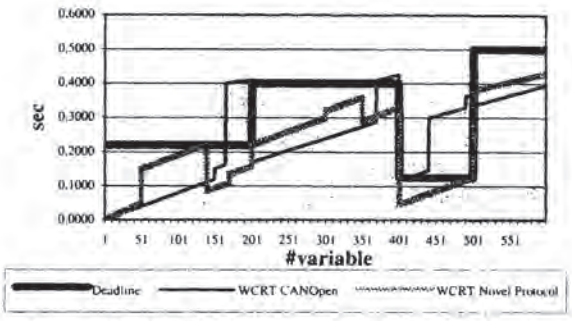

Figure 5 - Worst Case Response Time Evaluation

7. CONCLUSIONS

The paper has presented a protocol for dynamic assigning COB-IDs in a CANOpen communication system. The assignment strategy is an original one and is perfectly compatible with the Application Layer standard, as shown in the paper.

REFERENCES

Avind K., Ramamritham K. and Stankovic J.A. (1991). A Local Area Network Architecture for Communication in Distributed Real-Time Systems. Real-Time Systems. Vol.3, No.2.

Cavalieri S. (2001). A Protocol for Dynamic Assignment of Identifiers in Can Application Layer. Proceedings 4th IFAC FeT Conference Fieldbus Technology, Nancy, 15 - 16 November 2001, Palais des Congrès, pp.89-96.

Cenelec, EN50325-4 (2001). Industrial Communications Subsystem based on ISO 11898 (CAN) for Controller-device Interfaces. Part 4: CANOpen.

CiA/DS201. (1996). CAN Application Layer for Industrial Applications: *CAN in the OSI Reference Model*.

CiA/DS204-1. (1996). CAN Application Layer for Industrial Applications: *DBT Service Specification*.

CiA/DS204-2. (1996). CAN Application Layer for Industrial Applications: *DBT Protocol Specification*.

CiA/DS301. (1996). CAL-based Communication Profile for Industrial Systems-CANOpen. Version 3.0.

ISO IS-11898. (1993). Road Vehicle-Interchange of Digital Information-Controller Area Network (CAN) for High Speed Communication.

ISO 9074. (1997). Estelle: A Formal Description Technique based on an Extended State Transition Model.

Ouni, S., Kamoun, F. (1999). An Efficient Protocol for Hard Real-Time Communication on Controller Area Network. Proceedings 6th international CAN Conference '99. pp.04-16, 04-23.

Tindell K.W., Hansson H. and Wellings A.J. (1995). Calculating Controller Area Network (CAN) Messages Response Times. Control Engineering Practice. Vol.3, No.8.

Tindell K.W., Hansson H. and Wellings A.J. (1994). Analysing Real-Time Communications: Controller Area Network (CAN). Proceedings Real-Time Systems Symposium. pp.259-263.

Tovar E. (1999). Supporting Real-Time Communications with Standard Factory-Floor Networks. Ph.D. Thesis, University of Porto.