DESIGN AND IMPLEMENTATION OF A CAN / ETHERNET BRIDGE

H.Ekiz, A.Kutlu, M.D.Baba, E.T.Powner
University of Sussex, Brighton
BN1 9QT, United Kingdom

Abstract

The Controller Area Network (CAN) is a serial bus with high speed, high reliability, and low cost for distributed real time control applications in both the car and the industrial environments. In an industrial environment, while the CAN is used by manufacturing sections to control the systems, the management department can use Ethernet (or Token Ring) LAN in its building. This implies that the CAN should communicate with a LAN. A solution is to use the internetworking devices to connect a CAN and a LAN.

Bridges are high performance devices that are used to interconnect two similar or dissimilar LANs. The aim of this study is to design and implement a bridge which is capable of connecting a CAN and an Ethernet LAN.

In the following, a brief overview about the CAN, the Ethernet, and bridges is presented. The required processes are detailed and the implementation of the bridge is explained in section 2 and 3, respectivelly. In section 4, the modelling environment for simulation is summarized. Finally, the results obtained from simulation are presented in section 5, and the conclusion is given in section 6.

1. Introduction

The fast growing use of the Controller Area Network (CAN) in many industrial applications soon resulted in a specific CAN-based networks. In CAN-based distributed control systems, two of the potential problems are: the size of the distributed area and the communication with other LANs (e.g., Ethernet or Token-ring). In the first case, the appropriate solution is to divide the network into segments and connect them using a repeater or a MAC bridge [1]. In an industrial plant, whilst the manufacturing unit uses CAN, the Ethernet can be employed by the management department. In this case, the problem is communication between two different LANs. One of the solutions is to use a bridge which is capable of connecting a CAN and an Ethernet LAN. This paper is concerned with the design and implementation of a 'CAN / Ethernet Bridge', Using such an approach, it is necessary to consider three subjects in some detail: the CAN, the Ethernet and the bridge.

1.1. The CAN and The Ethernet

The Controller Area Network (CAN) is a high performance and highly reliable advanced serial communication protocol which efficiently supports distributed real-time control at high speed, low cost, and a very high level data security. The Ethernet is one of the most popular LANs in technical and office environments [2]. It operates at a data rate of 10 Mb/s and works equally well over twisted-pair, coaxial cable, and fiber.

Both the CAN and the Ethernet systems have different frame formats (Figure 1) and practise different Medium Access Control (MAC) mechanisms and routing algorithms. For routing, the CAN uses the selection mechanism (algorithm) and the Ethernet practises the address routing mechanism. As the MAC mechanism, the CAN protocol uses the Collision Sense Multiple Access / Collision Resolution (CSMA/CR), whilst Collision Sense Multiple Access / Collision Deduction (CSMA/CD) is used by the Ethernet protocol.

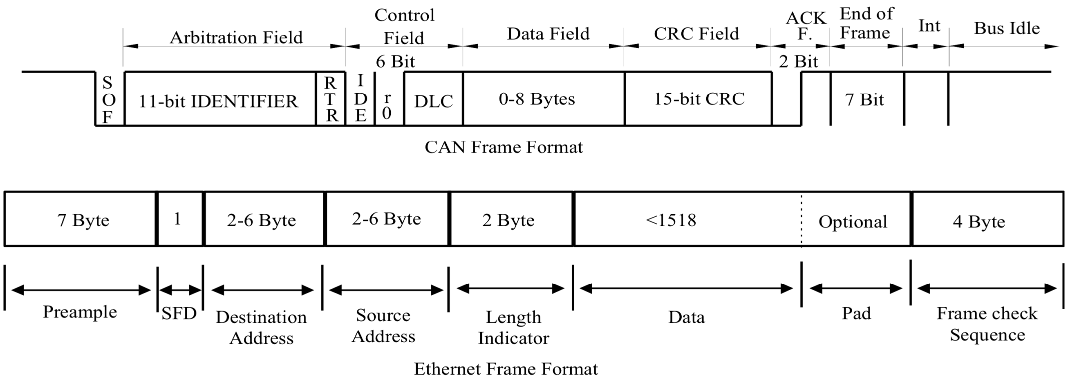

Figure 1. CAN and Ethernet frame formats

1.2. Bridges

Bridges are high performance devices that are used to interconnect similar or dissimilar LANs. The interconnection of the dissimilar LANs (e.g., a Token Ring / Ethernet or a Ethernet / CAN) is more complicated than the interconnection of the similar LANs (e.g., a Ethernet/Ethernet or a CAN/CAN) in terms of process to be performed. Whilst the pass-through forwarding process is sufficient for the interconnection of the similar LANs, both the translation and forwarding processes are required for the interconnection of dissimilar LANs. Both types of forwarding processes, pass-through and translation, are performed by the bridges that operate at the Logical Link Control (LLC) or Medium Access Control (MAC) level [3]. The MAC level bridging provides a connection between LANs that use the same frame format and employ the same data link protocol. The LLC level bridging provides the unique situation of enabling bridges to achieve the interconnection of LANs that use dissimilar frame format and different data link layer protocol [3] [4]. This implies that translation is necessary when the LANs have different frame formats.

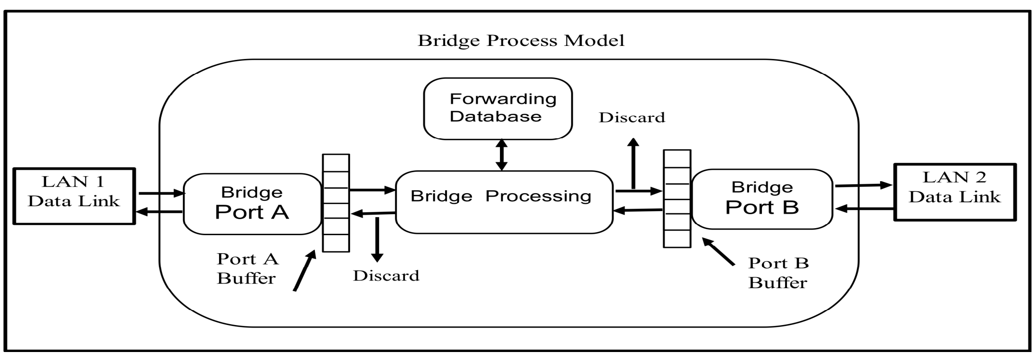

Figure 2. Abstract bridge implementation

In a bridge, information that is either stored at the bridge or provided within the transmitted frame assists the bridge in making a simple decision for routing: pass the frame to the next segment (known as forwarding) or not pass the frame (known as discarding) (Figure 2). The decision to forward the frame to its intended destination is made by consulting the forwarding database in the bridge that contains the destination addresses. If the destination address is found in the forwarding database, the frame is sent to that destination. If the destination address is not found, the discarding process is applied. The only exception is the learning process. This process is executed at the times when the bridge is first plugged in and at other predifined periods. When the bridge is first plugged in, every incoming frame is transmitted to the other side of the bridge. This action is referred to as flooding [2]. As time goes on, the bridge learns the addresses of all stations interconnected via that bridge.

1.3. Designing a CAN / Ethernet Bridge

The bridge to be designed will be a two port 'CAN / Ethernet Bridge' which is capable of connecting a CAN and an Ethernet systems. The bridge contains the worst case translation that requires creation or loss of fields representing unmatched services. For example, the CAN supports priority but the Ethernet does not. In this case, the translation process loses the priority. When forwarding in the opposite direction, the bridge must insert the priority. Another incompatibility is in frame sizes. The Ethernet supports a larger frame size than the CAN. Therefore, translation requires the adding or removing of padding.

The structure of the bridge may be different from the implementation point of view. However, in general, the processes which should be performed on the frames and the desired services in a CAN / Ethernet bridge will be the same. The number of bridge elements, their domain of operations, and their relations to each other should be such that they are able to perform the processes required and provide the necessary services which are going to be detailed in the following sections. The CAN / Ethernet Bridge (Figure 3) will be explained under two topics: the functionality (design principles) and the implementation.

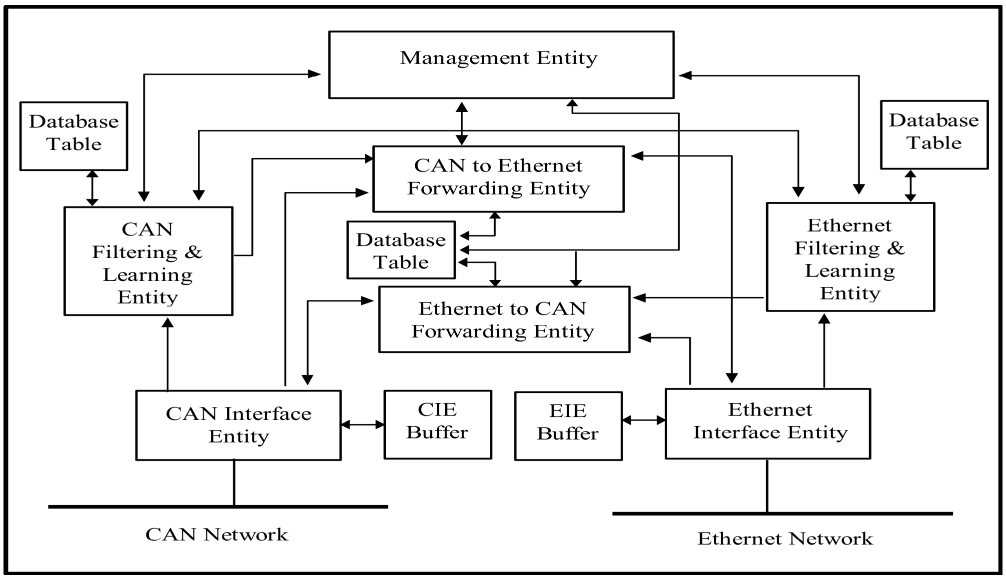

Figure 3. The functionality diagram of the CAN to Ethernet bridge

2. Modelling the Functionality of the Bridge

The functionality of the bridge is going to be detailed in two topics: Frame Reception / Transmission and the Bridge Management. As stated, each port of the bridge has different protocol, frame format, and frame reception / transmission mechanisms. This results that the processes to be performed by each port of the bridge will be different.

2.1. The Reception and Transmission of the Frames

The reception and the mapping processes in the bridge will be explained for each port separately

2.1.1. Receiving the CAN Frames and Reformatting into the Ethernet message format

The receiving process on the CAN side is performed by the 'CAN Interface Entity' (CIE). When the CIE receives a CAN frame without an error, it completely transfers it into a buffer and informs the 'CAN Learning and Filtering Entity' (CLFE). The CLFE performs two functions: to create the database tables during the learning process and to perform filtering (discarding/forwarding) process using these database tables. The creation of the database tables and their location is explained in section 2.1.3.

During the filtering process if the CLFE decides to forward the received frame, it passes the frame to the 'CAN to Ethernet Forwarding Entity' (CEFE). The CEFE reformats the CAN frame into the Ethernet frame format. Lastly, the reformatted frame is transferred to the 'Ethernet Interface Entity' (EIE) for retransmission to the Ethernet side of the bridge.

The reformation from the CAN frame into the Ethernet message format is performed in three phases:

i-Discard unnecessary parts of the frame: SOF, ACK, End of Frame fields do not exist and , the value of CRC field is not valid any more in the Ethernet message. Therefore, these fields are discarded.

ii-Modify invalid parts of frame: FCS, Control Field (DLC), Data Field, Arbitration field (assumed as source address) field of CAN frame should be modified depending on the new message format.

Depending on the implementation, all or the DLC part of the Control field (6 or 4 bits length) is modified as the Length Indicator Field of new message (16 Bits length).

The MSB of each data byte is transmitted first in the CAN while the LSB of each data byte is transmitted first in the Ethernet. So, the order of each byte of the received data field should be modified (reversed) based on the new frame format.

The CAN frame does not have the address information field. Therefore the bridge is responsible for incorporating the address fields into the Ethernet message format. To provide this service the Arbitration field is modified as a source address. But the arbitration field of the CAN frame has 11 bits against 6 bytes of destination address field of the Ethernet frame. Therefore, 11 bits value is changed 6 bytes value by adding new bits, '0s'. Moreover the destination address is created by using arbitration field and database tables.

iii-Add new parts: Preamble, SFD, FCS, Pad fields do not exist in the CAN frame format. These fields should be incorporated by bridge in the new message.

The FCS field, four-byte, should be computed and appended to the new frame based on contents of destination address, source address, length, data, and pad fields. As mentioned before, the CAN frame can have maximum 8 octets data field. But, at least, a 46 octet data fields should be provided for the Ethernet frame format. Therefore an integer number of padding bytes, depending on the length of the data field of the CAN frame, should be added to the data field of CAN frame as pad.

2.1.2. Receiving messages from the Ethernet side and Reformatting into the CAN frame format

The receiving process in the Ethernet side is performed by the EIE. The EIE receives all messages from the Ethernet bus and performs pre-filtering process checking I/G bit of the destination address (the first bit of the destination address field). According to the value of I/G bit, the frames received from the Ethernet side are either discarded or buffered (copied) into a memory (EIEB) by the EIE.

Normally, If 'I/G' bit of the destination address field of the Ethernet frame is '0-zero', this means that the frame is addressed to a specific node. It is proposed that if the 'I/G' bit is set to '0', this indicates that the destinated node is located in the Ethernet side of the bridged system. When the bridge receives this type of frame, it immediately discards the frame without using the buffering and searching process. This results in the bridge gaining time and buffer space. If 'I/G' bit is '1', this indicates that the frame is addressed to either a node in the CAN side or a group of nodes in the Ethernet side. These kind of frames are buffered and reported to the 'Ethernet Learning and Filtering Entity' (ELFE) by the EIE. Then the second (main) filtering process is performed by ELFE.

When the ELFE is informed that the bridge has received a message from the Ethernet side, it searches the database table which contains the addresses of the nodes located in the Ethernet side. If the destination address(es) of the received frame is found in the database table, the received frame is discarded. If the destination address(es) is not found, the message is forwarded. The message to be forwarded is reported to the 'Ethernet to CAN Forwarding Entity' (ECFE) for the mapping process.

The frames which are reported to the ECFE are reformatted by the ECFE in three steps:

i-Discard unnecessary parts of the frame: Preamble and pad fields are not present and the FCS field is not valid any more in the CAN frame. Therefore preamble, pad, and FCS fields are discarded.

ii-Modify invalid parts of frame: SFD, Destination address, Source Address, Length indicator, Data fields of the Ethernet frame should be modified depending on the CAN frame format.

It is proposed that a frame sent to the CAN system can have a maximum data field of 8 octets. Therefore a four bit length is quite enough to represent the length of the data field. So, the last four bits of the length indicator field are used to modify the control field of CAN frame.

The order of each byte of the data field should be reversed based on the new frame format. In addition, the SDF field is changed to one dominant bit (SOF) for adapting to the CAN frame format.

It is proposed that any node in the Ethernet system sends an eight bytes length data and this data is increased to 46 bytes length by adding a corresponding pad field. Therefore, the first 8 bytes of the data is used to modify new data format and rest is omitted.

It is proposed to use the destination address of the Ethernet frame as the arbitration field of the CAN frame. When a node is sending a message to the CAN side, it puts the arbitration field value of the CAN frame into the destination address field of the Ethernet message. The proposed method will not need to use a database table for the incorporation of the arbitration field during the mapping.

iii-Add new parts: CRC, ACK, End of frame fields do not exist in the Ethernet frame format. Therefore these fields should be incorporated in the new frame format. The value of CRC, the ACK and End of frame fields are added by the CIE. The reformatted frame is passed to the CAN system.

During the all explanations above, it is assumed that the database tables were used in the both processes: in the filtering process and in the reformatting. To clarify the functionality of the bridge, the creation and the location of the database tables are explained based on the learning process.

2.1.3.The creation of the database tables and the learning process

To perform the filtering process and to provide the reformatting from the CAN frame format into the Ethernet message format or vice versa, the database tables are used. When the CAN and the Ethernet are connected using a bridge, the bridge does not have any information (database table) to perform both processes; the filtering and the reformatting. The database tables are created by the decision of one side of the bridge or according to the result of correspondence between the two ports during the learning process. Each side of the bridge creates its related database tables. So, the learning process and the creation of the database tables must be considered on both sides of the bridge.

The learning process and the creation of database tables of the CAN side:

During the learning process, the bridge learns which messages sent by CAN side are received by the Ethernet side. The learning process and the creation of database tables are summarised below:

i-It is proposed that each CAN message is sent at least once by related node in the CAN system, when the bridged network system is turned 'ON' and at predifined interval times. The CIE receives these CAN messages and buffers them. This process seems as if the bridge needs a large memory. However, the length of the CAN frame is short and the buffers which will be used to record the database tables is empty during the learning process. Hence, the bridge will not need extra large memories.

After all CAN messages have been stored in the memories, the flooding process is started from the CAN side to the Ethernet side. This means that the stored CAN frames are transmitted to the Ethernet side one by one. During the flooding process, the CAN frames are mapped into the Ethernet message format. In the mapping process, the incorporation of the address fields seems a big problem. To create address fields of the Ethernet frame, the arbitration field of the CAN frame (the one which is being processed) is used as a source address of the Ethernet frame, while one of the addresses in the database table (DBT1-contains all addresses in the Ethernet side) is used as the destination address. The creation of DBT1 is explained as follows. Here, the problem is the matching between the source and the destination address pairs. The solution is to map the address fields of the Ethernet frame as following: The bridge puts the arbitration field of the received CAN frame as the source address of the Ethernet message and all addresses of nodes in the DBT1 as the destination address of the Ethernet message (the broadcasting). After the required parts are incorporated, the mapped message is sent to the Ethernet side by EIE. Here, the problems are: how an Ethernet node knows that the message is belongs to its network and how the bridge recognises that the message is received by a node(s).

ii-The solutions are: any node in the Ethernet side sends a reply message when it receives a appropriate message from the bridge during the learning process. This means that when the EIE sends a message, it waits for one or more reply messages. If it does not receive any reply message in a predefined period, it assumes that the message was not accepted by any node on the Ethernet side. This implies that the arbitration field of the CAN message is not added to the database tables.

In this step, the question is that how a node on the Ethernet side decides on whether the incoming message should be received or not. The nodes decide by checking the source address of the coming message. If it finds that the source address of the incoming message is known, it receives the incoming message and sends a reply message to the bridge. If one or more reply message are received by the EIE, the entity reports this to the CEFE and the CEFE records the addresses pairs to a database table (called as DBT 2): i-The arbitration field of the CAN message as the source address. ii-The source address of the reply message(s) as the destination address(es).

iii- The CEFE records the address pairs and reports to the CLFE. The CLFE records this data its own database table (DBT 3). Thus, table 3 will contains the arbitration fields of the CAN frames to be forwarded to the Ethernet side and will be used during the CAN filtering process.

The learning process and the creation of database tables in the Ethernet side:

During the learning process, the bridge learns that which messages sent by Ethernet nodes were received by the CAN side. For this to happen:

i-It is proposed that each node in the Ethernet side sends a message to the bridge during the learning process and at predifined interval times. As known, the source address field of the messages indicates the addresses of the nodes which are located in the Ethernet side of the bridge. The source addresses of the received messages are extracted and written into a memory (DBT 1). The DBT1 are also used by the ECFE as explained previously.

ii- Every message from the Ethernet side is reformatted (mapped) into CAN frame format and forwarded to the CAN side (flooding process). It is proposed that the destination address field of the Ethernet message contains the arbitration field of the CAN frame, if the message destination is on the CAN side. The 11 LSBs of the destination address field is used to incorporate the arbitration field.

The frame mapped in the CAN frame format is sent to the CAN side by the CIE. If the CIE acknowledges that a node(s) received the frame, the CIE reports this to the ELFE and the ELFE records the destination address of the frame into a database table (DBT1). Thus it is necessary to consider the CAN protocol features. In CAN, when a node transmits a frame, it monitors the bus to ascertain if the frame is being received by a CAN node(s). If it does not monitor the changing of the acknowledge field, it assumes a error condition and retransmits the frame again. The solution of this problem is explained in [4].

2.2. Bridge Management (BM)

The BM performs four functions to increase the throughput of the bridge, summarised as follows:

2.2.1. Routing Algorithm Management: Ethernet uses CSMA/CD as the MAC access method therefore the most appropriate method for routing is either the spanning tree algorithm or the self-learning process. If it is assumed that the bridged system has simple topology, the learning process will be enough for routing management in the bridge.

2.2.2. Security management: The bridge can provide facilities to protect specified node(s) being accessed by unauthorised node(s) in the bridged system. For this purpose, a dedicated static table should be added to the involved database of ‘Filtering and Learning Entity’.

2.2.3. Fault management: The source of the error in the bridged system can be in the bridge, connected CAN, or the connected Ethernet. The fault management mechanism diagnoses the error(s) and tries to recovery. These errors may be because of hardware and/or software malfunctions [5], [6].

iv-performance management: A set of statistical counters can be used in the bridge and the bridged system to improve the bridge and the bridged system performance.

3. The Bridge Implementation and the Function of the Entities

The implementation of the CAN / Ethernet bridge is explained in two phases: the way that the processes are performed and the structure of each element (called as entity) in the bridge. The implementation of the CAN / Ethernet Bridge will have a general block diagram as shown in Figure *3*. The function of the each entity is summarised in following.

3.1. The CAN Interface Entity-CIE

The features of the CIE is the same as the other node's features connected to the CAN system. A CAN controller module which is produced by the I&Me company is used as the CIE [7]. The CIE receives and buffers all frames from the CAN system. The CIE not only provides a functional interface between the CAN system and the bridge but also both direct Input/Output (I/O) and direct memory access (DMA) to their attached boards.

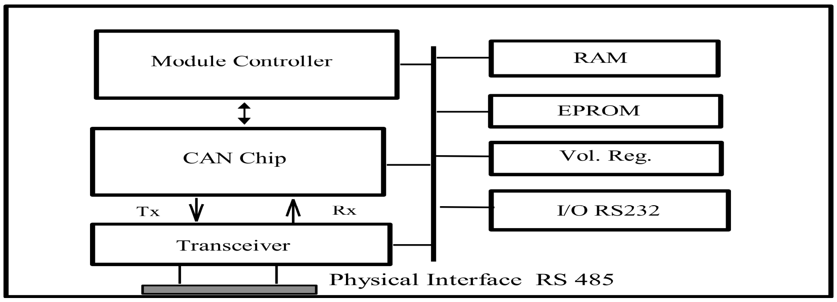

Figure 4.The Units of the CAN Interface Entity

3.2. The CAN Learning and Filtering Entity (CLFE)

The CLFE provides two services: to create the node address table of the CAN side (DBT3) during the learning process and to perform the filtering process. During the filtering process, the CLFE inspects the received frame in the DBT3 to determine whether the frame will be forwarded or discarded. If the arbitration field of the frame matches any arbitration field in the DBT3, it is reported to the CEFE. If the arbitration field is not in the DBT3, the frame is discarded.

Before the searching process in the DBT3, the arbitration field of the CAN frame buffered is extracted from the CAN frame. Since the length of the arbitration field is 11 bits (small) and the location of the arbitration field in the frame is easy, a small comparison time during the filtering process will be enough. The extraction of the arbitration field of the CAN frame and comparison of the arbitration fields, where one was extracted from received frame and other was read from database table are executed by the CLFE. The detail of these processes can be found in [4].

3.3. The CAN to Ethernet Forwarding Entity (CEFE)

The CEFE reformats the CAN frame into the Ethernet message format. After reformatting, the CEFE sends the frame to the EIE for transmission to the Ethernet side.

During the mapping process from the CAN frame to the Ethernet frame format, one of the important issues is the bit reordering performed by the CEFE. This process can be performed by software or by hardware using peripheral devices. If the reordering is performed in hardware, it may be performed without any delay. The details of this process can be found in [6].

3.4 Ethernet Interface Entity (EIE)

The EIE provides an interface between the bridge and the Ethernet network. It is proposed that the EIE is programmed to work in a destination addressing scheme based on the I/G bit of the destination address field (first bit). In this mode the entity performs a prefiltering process depending on the I/G bit.

It is proposed that the I/G bit of the destination address field of any message which is sent to CAN side should be '1'. This provides two benefit for the system: The bridge can easily check I/G bit and decide that if the message destination for the Ethernet side or not. If this bit is '0', the bridge understands that this message belongs to the Ethernet side and immediately discards the frame. If the I/G bit is '1', it then continuous the processes.

For the EIE, a Media Access Controller for the Ethernet (MACE) chip is used. The MACE is a 16-bit controller with a superior modular architecture and versatile system interface that allows it to be configured as a stand-alone device or as a connectivity cell incorporated into a larger, integrated system. Details about design of such an Ethernet Interface can be found in [8].

3.5. Ethernet Learning and Filtering Entity (ELFE)

The ELFE provides two services: to create the node address table of the Ethernet side (DBT1) and to perform the main filtering process. The main filtering process is performed in two steps.

In the first step, the destination address of the received frame and content of the DBT1 are compared. If the destination address (group address name) of the received frame are found in the DBT1, the ELFE discards the frame. If not, the second step of filtering process is performed.

In the second step of the filtering process, the destination address field of receiving frame is extracted and the ELFE compares the destination address of the received frame against the content of the DBT2. The DBT2 contains the destination addresses of the Ethernet messages which were received by one or more CAN nodes during the learning process. If the destination address of the frame matches with an arbitration field in the DBT2, the ECFE is informed, otherwise, the frame is discarded. Here, it can be asked as to why two steps of the filtering processes are performed. The answer is the ratio of Local / Remote messages. Most of the messages are local and thus the entity can finish the filtering process after a few comparison (takes very short time).

3.6. Ethernet to CAN Forwarding Entity (ECFE)

The ECFE reformats the Ethernet frames into the CAN frame format. It uses the destination address of the Ethernet frame to create the arbitration field of the CAN frame as explained previously. During the reformatting from the Ethernet to the CAN frame format, one of the important issues is the bit reordering. The ELFE performs the data byte reordering process as explained in the CEFE section. After reformatting, it sends the frame to the CIE for transmission into the CAN side.

3.7. Bridge Management Entity (BME)

The BME is dedicated to managing functions in order to increase the throughput of the bridge. This entity has capability to manage all other entities. The BME performs all functions mentioned in the 'Bridge Management' section. The functional block diagram of the BME is illustrated in figure 5. The central processing unit performs all control processes in the bridge and interactions with all the entities of the bridge. A dual universal receiver/transmitter (DUART) interfaces the entity to the output terminal through RS232 interface. Timers/Counters are used to assign the age of the frames.

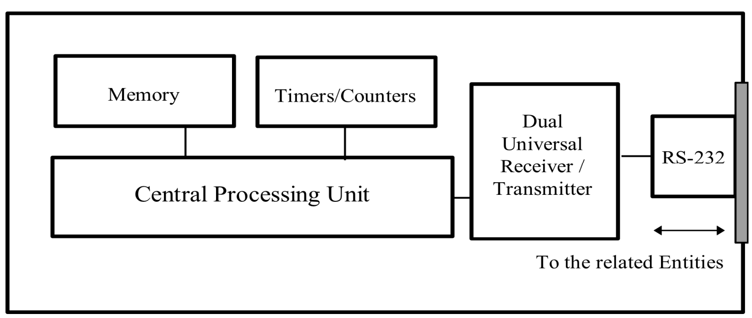

Figure 5. The Functional Block Diagram of the Bridge Management Entity

4. The Modelling Environment

A commercial simulation package that has been developed to model network systems and network devices is used to model the bridge and all systems shown in Figure 3. In the simulation, the program models three systems: the bridge elements, CAN network, and the Ethernet LAN. For simplicity, the simulation model for each system and each bridge entity is based on following assumptions.

i-The CAN and the Ethernet systems: The IEEE 802.3 stardard (10BASE-5) is used to model the all Ethernet features. In the implementation of the CAN system and the CIE, the CAN board features from the I&Me product were used. The CAN system bus speed was chosen as 1 Mbit/sec.

ii-The CAN / Ethernet Learning and Filtering Entities : The learning and filtering processes of each port of the bridge is done in parallel by each 'Learning and Filtering Entity'. A M68000 microprocessor with its peripherals is used to implement each of the Learning and Filtering Entities.

iii-The CEFE and ECFE modules: Each of the these entities comprises two part; a M68000 microprocessor based forwarding part and a memory. It is assumed that in order to manage the database tables of the CEFE and ECFE, Contents Addressable Data Managers (CADM) were used.

iv-The EIE : It is assumed that the MACE device is used as the EIE module [8].

v-The BME : A M68000 microprocessor with its peripherals was used to model the BME. The process of the DUART and the RS232 elements were defined as delays.

vi-As memory, to built up database tables, the Sony product memory features were chosen.

vii- In the networking system, the message traffic characteristic was defined as 30% remote messages and 70% is local messages.

5. Simulation Results

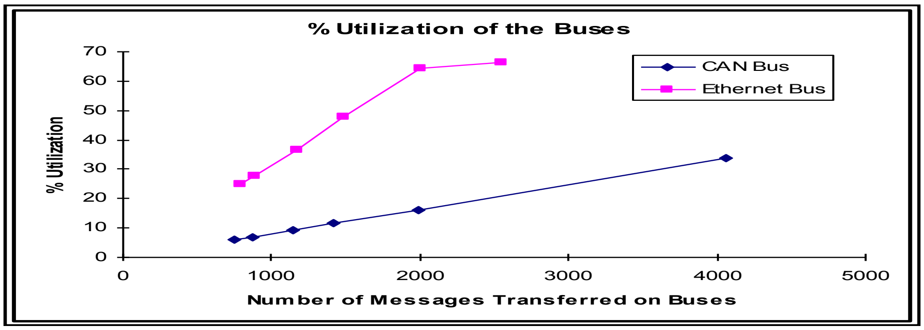

Figure 6. Utilisation of the buses in the system designed

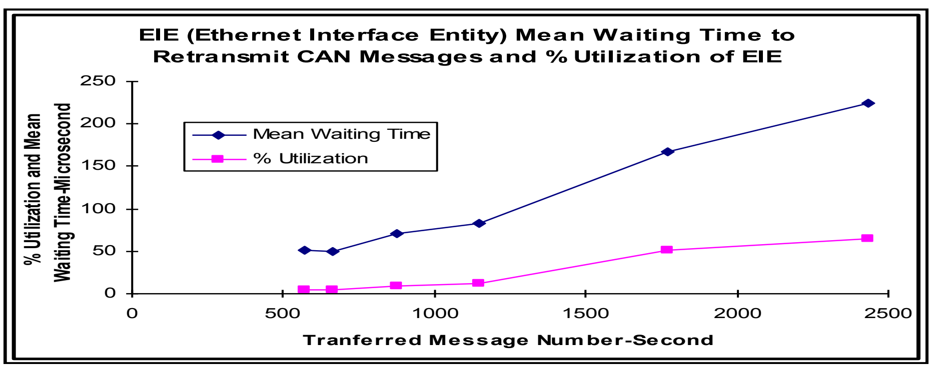

Figure 7. Utilisation and main waiting time of the EIE

To obtain the designed bridge performance, the model seen in Figure 3 was used. This model indicates that the CAN / Ethernet bridge not only provides a service to interconnect a CAN system and a Ethernet LAN but also performs all required function for the communication between end systems. The performance of the designed bridge was evaluated from the utilisation of the bus systems (Figure 6), the utilisation of the some bridge elements (Figure 7), and total processing time of the bridge statistics (Figure 8) with different loads. From the figures, it is concluded that both sides of the bridge and the both systems can support message traffic up to 50% utilisation of the buses. It can be deduced from the graphs shown in Figure 6 and Figure 7 that the Ethernet side message number should be less than 2000 messages / second to get enough performance from the both the bridge and the Ethernet LAN. Beyond this number of messages, the performance will be strongly influenced by the increase of the load. However, the system suports message traffic upto 5000 mes/sec on the CAN side.

The rate at which frames are processed and forwarded for transmission from one port to another is called the bridge forwarding rate. As can be seen in Figure 9, the bridge forwarding rate affects the process time of the messages only with the load more than 2000 messages per second. Another element affected by the load is buffer capacity. As the loading of the buses increases, the number of queued (buffered) message to be processed within the bridge increases. The results presented in

Figure 8 quantify the required buffer storage capacity, which is related to the number of queued message, for various network loads. Figure 9 shows the maximum waiting time of the CIE in the bridge to retransmission of the received frames with different priorities. It is concluded from the graphs that the CIE should have the highest priority in the CAN system to reduce the transmission delay.

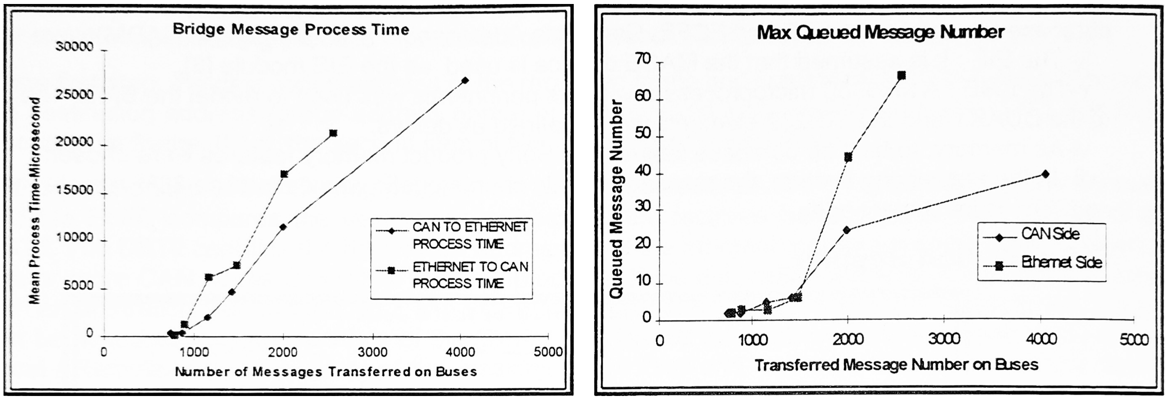

Figure 8. Total processing times and queued message number in both ports of the bridge

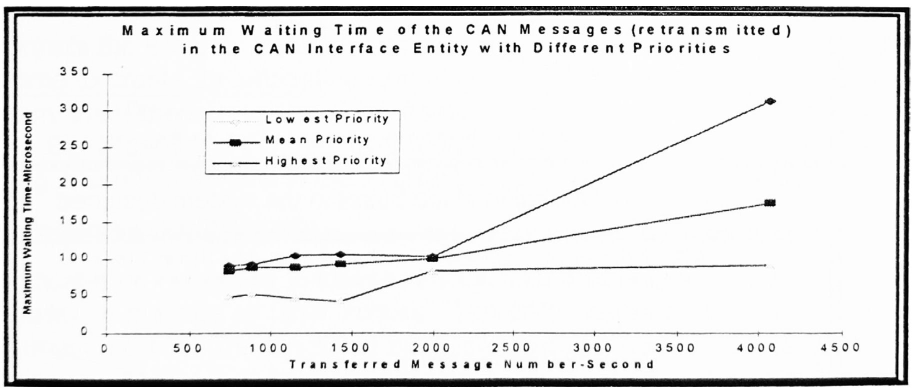

Figure 9. Maximum waiting time of the CIE for retransmission

6.Conclusion

A bridge must provide a selective frame retransmission function and interface operation which allows communication between dissimilar systems. The objective of the this research has been to design a bridge which provides a service to achieve the interconnection of the CAN and the Ethernet. The designed bridge has fulfilled the objectives and incorporates the translation and filtering processes. In summary, the operational model of the bridge includes four phase of operation. First, the bridge receives a frame from one of the CAN or the Ethernet. Second, the bridge decides whether or not to forward the frame. Third, the bridge reformats the frame to the required format. Lastly, the bridge transmits the frame to the other system.

The designed bridge is a transparent bridge and uses the learning process to built-up the forwarding database.The parameters, process time, retransmission delay, and required buffer size which are related to the performance of the bridge are satisfactory in meeting the overall requirements.

References:

[1]. Ekiz H., Powner, ET., Kutlu A. Performance Analysis of CAN in Bridged Systems, *Proc. of II Communication Conference*, Manchester, June 1995

[2]. Halsall F*, Data communication, computer networks, open systems*, Addison-Wesley Pub. 1992, USA.

[3]. Hawe, B., Kirby, A., Transparent Interconnection of Local Area Network with bridges, *Telecomminication Network* vol 3, no 2, pp 116-130, 1984

[4]. Ekiz H., Kutlu A., Design and Implementation of a CAN / CAN Bridge, *Proc of IEEE I-SPAN 96*, June 1996, Beijing, China.

[5]. IEE Standard 802.1: Pard D, MAC Bridges, 1991, *IEE Computer society*.

[6]. Haghichi, G., Ethernet/Token Ring Bridge Design, *PhD thesis*, Salford-1990.

[7] BCAN Network Evaluation System with 80C552/80C196 CPU User Manuel, I&Me, 1990, Germany.

[8]. EThernet/IEEE 802.3 Family, Am79C940, *Advanced Micro Device Handbook*, 1994, USA.