DESIGN AND IMPLEMENTATION OF A CAN/ATM LAN BRIDGE

İbrahim Özçelik[1] İsmail Ertürk[2] Hüseyin Ekiz[3] Füsun Selçuk[1]
e-mail: ozcelik@sakarya.edu.tr erturk@kou.edu.tr ekiz@sakarya.edu.tr fusuns@sakarya.edu.tr
[1]*Sakarya University, Faculty of Engineering, Department of Computer Engineering, 54050, Esentepe, Sakarya, Turkey*
[2] *Kocaeli University, Faculty of Technical Education, 41300, İzmit, Kocaeli, Turkey*
[3]*Sakarya University, Faculty of Technical Education, 54050, Esentepe, Sakarya, Turkey*

Keywords: Internetworking, Bridge, Controller Area Network, Asynchronous Transfer Mode

ABSTRACT

The Controller Area Network (CAN) was developed to support the communication link between remote subsystems in automotive industry, and it has become a desirable, cheap solution for networking in industrial environments [1]. However, problems arise when the CAN is used in industrial environments since there is a limit on the maximum length of a single CAN bus and when the CAN systems need to communicate with existing networks. ATM LAN is a recently developed technology. It is believed to be the ideal solution for B-ISDN. This type of network is so versatile that can be used in any kind of environment. Therefore the need of a CAN to communicate with an ATM network will be a fact in the near future. Since the CAN message carries no information related to the destination and source addresses, it is not possible to use traditional address based internetworking devices to connect CAN segments, or CAN and existing LANs such as Ethernet and ATM [2, 3]. This results in a fact that a new bridge based on CAN and ATM features should be designed. This paper is concerned with the design and implementation of a 'CAN / ATM Bridge'.

I. INTRODUCTION

In an industrial environment, while the CAN is used by manufacturing sections to control the systems, the management department can use a ATM LAN. In addition, since the CAN bus has a limited length, the CAN-based industrial application can need a backbone system, such as ATM, to extend the size of the distributed area. This implies that the CAN and ATM should communicate with each other.

The CAN frames and the ATM cell have different formats (Figure 1) and practise different routing algorithms. For routing, the CAN uses connectionless method while the ATM has connection oriented routing mechanisms. In addition, the CAN uses a shared bus with medium access control (MAC), while a star topology solution is preferred in ATM LANs.

A method to connect CAN and ATM is to use one of the internetworking devices: Bridge, Router, etc. Bridges are high performance internetworking devices that are used to interconnect LANs. Using such an approach in our work, it is necessary to consider three subjects: the CAN, the ATM and the Bridge.

II. THE CAN, THE ATM, AND THE BRIDGE

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 CAN data frame, Figure 1, contains information to synchronise, identify, control and save the data flow together with the user data. The CAN, despite serial in nature, is unlike many serial communication protocols; it does not contain destination or source address information. Instead, the message with an identifier indicates the type of information carried. The identifier is not only used to identify the message but also used in the arbitration mechanism.

ATM is considered as a specific packet oriented transfer mode based on asynchronous time division multiplexing and the use of fixed length cells. Each cell consists of an information field and a header. The header is primarily used to identify the cells belonging to the same virtual channel within the asynchronous time division multiplex, and to perform the appropriate routing. Cell sequence integrity is preserved per virtual channel.

Bridges are high performance devices that are used to interconnect similar or dissimilar LANs [4]. 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. In a bridge, information that is either stored 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 (forwarding) or not pass the frame (discarding). The only exception is the learning process. This process is executed when the bridge is first plugged in and at other predefined 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 [5]. As time goes on, the bridge learns the addresses of all stations interconnected via that bridge and destination of the frames that should be forwarded.

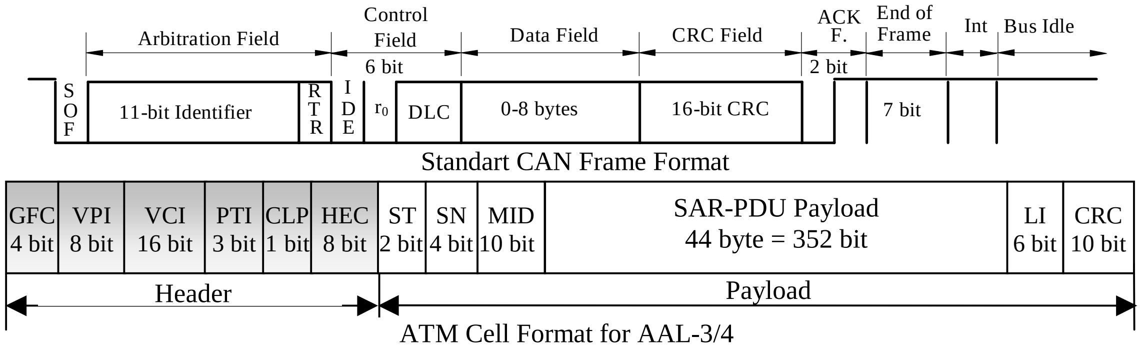

Figure 1: CAN frame and ATM cell structures

III. DESIGNING AND MODELLING A CAN / ATM BRIDGE

The bridge to be designed will be a two port 'CAN / ATM Bridge' which is capable of connecting a CAN and an ATM LAN (Figure 2). The bridge contains the worst case translation that requires creation or loss of fields representing unmatched services. For example, one of the incompatibles of this translation is in frame/cell sizes. The ATM supports a larger frame size than the CAN does. Therefore, translation requires the adding or removing of padding. The structure of the bridge may be different from the design and implementation point of view. However, in general, the processes which should be performed on the frames and the desired services in a CAN / ATM 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 will be able to perform the processes required and provide the necessary services which is detailed in the following subsections.

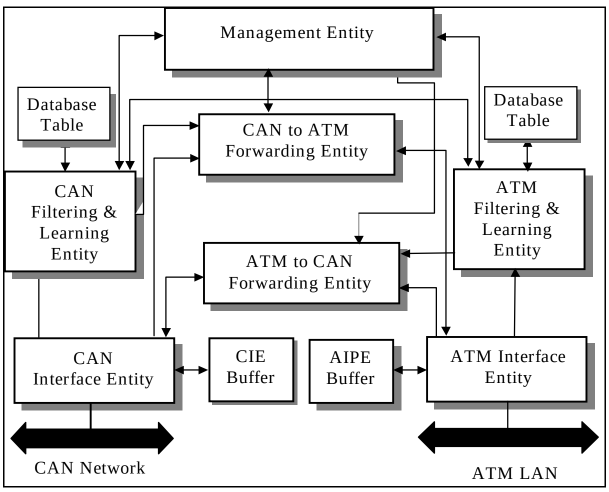

Figure 2: The functionality diagram of the bridge

As stated, each port of the bridge has different protocol, frame format, and frame reception/transmission mechanisms. Therefore the processes to be performed by each port of the bridge will be different.

CAN TO ATM TRANSFERRING PROCESS

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 a database table during the learning process and to perform filtering (discarding/forwarding) process using this database table.

During the filtering process, if the CLFE decides to forward the received frame, it passes the frame to the 'CAN to ATM Forwarding Entity' (CAFE). The CAFE provides a service to create the necessary parts for mapping the ATM cell format. Lastly, the created parts are transferred to the 'ATM Process Interface Entity' (APIE) for transmission to the ATM side of the bridge as ATM cell.

- Discard unnecessary parts: SOF, IDE and r0 bits in the Control field, ACK, End of Frame fields do not exist and, the value of CRC field is not valid anymore in the ATM cell. Therefore, these fields are discarded.
- Modify invalid parts of frame: Arbitration Field, DLC in the Control Field and Data Field of CAN frame should be modified depending on the new message format.
- Add new parts: ATM Cell Header Fields (GFC, VPI/VCI, PTI, CLP, Header Checksum), and ST, LI and CRC fields do not exist in the CAN frame format. These fields should be incorporated by the bridge.

ATM TO CAN TRANSFERRING PROCESS

It is proposed to use the AAL3/4 connection oriented transmission protocol for the communication in the ATM network. The MID field of the ATM cell would then be translated 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 MID field. The proposed method will not need to use a database table for the incorporation of the arbitration field during the mapping process. In addition, it is proposed that if an ATM message is sent to CAN side, it contains only one cell.

The receiving process in the ATM side is performed by the APIE. The AIPE receives ATM cells from the ATM switch, and buffers into APIE buffer. When an ATM cell is buffered, 'ATM Learning and Filtering Entity' (ALFE) is informed. It then searches the database table which contains the MID fields of the cells to be forwarded to the CAN side. If the MID field of the received ATM cell is found in the database table, this means that the cell is to be forwarded to the CAN side. The message to be forwarded is also reported to the 'ATM to CAN Forwarding Entity' (ACFE) for the mapping process. The messages which are reported to the ACFE are reformatted (mapped) in three steps by the ACFE and CIE:

1. Discard unnecessary parts of the frame: ATM Cell Header Fields (GFC, VPI/VCI, PTI, CLP, Header Checksum), ST, LI and CRC fields are not present in the CAN frame. Therefore these fields are discarded.
2. Modify invalid parts of the frame: Payload of the ATM cell should be modified depending on the CAN frame format. It is proposed that any host in the ATM system sends an eight byte length data and this data is increased to 48 byte length by adding a corresponding pad field. Therefore, the first 8 bytes of the data are used to modify new data format and the rest of it is omitted. In addition, the MID and the SN fields are extracted to be used as arbitration field and DLC in the control field of the CAN frame, respectively.
3. Add new parts: SOF, CRC, ACK, End of Frame fields does not exist in the ATM cell. Therefore these fields should be incorporated in the new frame format. These fields are added by the CIE.

IV. THE CREATION OF THE DATABASE TABLES AND THE LEARNING PROCESS

The database tables are used to perform the filtering process and to provide the reformatting from the CAN frame format into the ATM cell format or vice versa. When the CAN and the ATM systems 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 relevant database tables.

V. THE BRIDGE IMPLEMENTATION AND THE FUNCTION OF THE ENTITIES

The implementation of the CAN/ATM 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/ATM Bridge will have a general block diagram as shown in Figure 2. The function of the each entity is summarised in following. More information about the entities which are related to CAN side, CIE and CLFE, can be found in [2] and [3].

THE CAN INTERFACE ENTITY

The features of the CIE are 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 [6]. 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 supports direct input / output (I/O) and direct memory access (DMA) to its attached boards. Details can be found in [6].

THE CAN LEARNING AND FILTERING ENTITY

The CLFE provides two services: to create arbitration fields table for the CAN side (DBT1) during the learning process and to perform the filtering process. During the filtering process, the CLFE inspects the received frame in the DBT1 to determine whether the frame will be forwarded or discarded. If the arbitration field of the received frame matches any arbitration field in the DBT1, it is reported to the CAFE. If the arbitration field is not in the DBT1, the frame is discarded.

THE CAN TO ATM FORWARDING ENTITY

The CAFE reformats the CAN frame into the ATM cell format. After reformatting, the CAFE sends the cell to the APIE for transmission to the ATM side. During the mapping process from the CAN frame to the ATM cell format, one of the important issues is to adjust the length of the payload field. Since the CAN data field can be maximum 8 bytes, the whole data field is extended for 44 bytes payload field by adding padding bits.

ATM PROCESS INTERFACE ENTITY

The APIE provides an interface between the bridge and the ATM switch. For the APIE, a Cisco Systems ATM Interface Processor (AIP) is used. The AIP is ideal for a wide range of ATM applications, supporting many features and standards designed for ATM and internetworking in general [7]. The AIP design includes ATM segmentation and reassembly chipset produced National Semiconductor (the FRED chipset). This chipset can simultaneously support AAL5 and AAL3/4. The AIP uses a pool of local memory to store packets during the segmentation or reassembly for maximum performance. Details about AIP can be found in [7].

ATM LEARNING AND FILTERING ENTITY

The ALFE provides two services: to create the MID fields table of the ATM side (DBT2) and to perform the filtering process in the ATM side of the system. In the filtering process, the MID field of the cell being received extracted and the ALFE compares the MID field of the received cell against the content of the DBT2. The DBT2 contains the MID fields of the ATM cells which were received by one or more CAN nodes during the learning process.

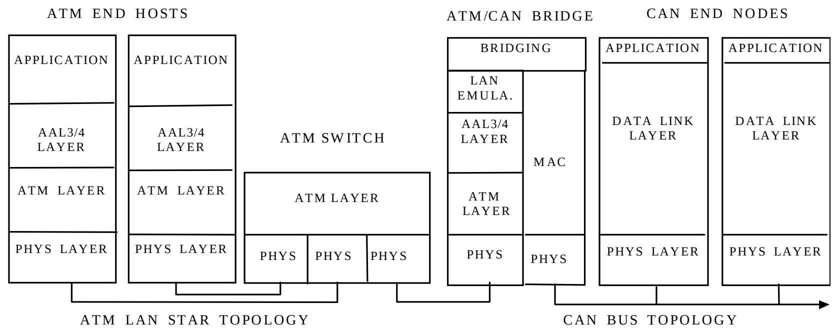

Figure 3. The layered system architecture of the CAN / ATM bridge

If the MID field of the received cell matches with any value in the DBT2, the ACFE is informed; otherwise, the frame is discarded.

ATM TO CAN FORWARDING ENTITY

The ACFE reformats ATM cells into CAN frames. It uses the MID fields of the ATM cells to create the arbitration fields of the CAN frames as explained previously. During the reformatting from the ATM to the CAN frame format, one of the important issues is the length of the data field. Maximum 8 bytes of the cell payloads are extracted (i.e. the following 36 bytes are omitted), and incorporated as data field of the CAN frame. After these processes, the ACFE sends the frame to the CIE for transmission into the CAN side.

BRIDGE MANAGEMENT ENTITY – BME

The BME performs four functions to increase the throughput of the bridge: Routing Algorithm Management, Security Management, Fault Management, and Performance Management. The BME is dedicated to perform functions mentioned above in order to increase the throughput of the bridge. In addition, this entity has capability of managing all other entities. Details of these processes and the BME can be found in [3].

VI. THE MODELLING ENVIRONMENT

A commercial simulation package developed to model network systems and network devices is used to model the bridge and all systems shown in Figure 2. In the simulation, the program models three systems (Figure 3): the bridge elements, the CAN network, and the ATM LAN. For simplicity, the simulation model for each system and each bridge entity is based on following assumptions:

1. The CAN and the ATM systems: In the implementation of the CAN system and the CIE, the CAN board features from the I&Me product were used [6]. The CAN system bus speed was chosen as 1 Mbit/s. ATM LAN speed is chosen as 155 MBit/s.
2. The Learning and Filtering Entities: The learning and filtering processes of each port of the bridge are realised 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.
3. The CAFE and ACFE modules: Each of these entities consists of two parts; a M68000 microprocessor based forwarding unit and a memory. It is assumed that in order to manage the database tables of the CAFE and ACFE, Contents Addressable Data Managers (CADM) were used.
4. The APIE: It is assumed that the Cisco Systems ATM Interface Processor is used as the APIE module [7].
5. The BME and Memory: A M68000 microprocessor with its peripherals was used to model the BEM. The process of the DUART and the RS232 elements were defined as delays. The Sony product memory features were chosen for memory to build up database tables.
6. In the networking system, the message traffic characteristic was defined with various local message /remote message ratio as %30/70, %50/50, %70/30 in CAN side, as %10/90, %20/80, %30/70 in ATM side, respectively**.**

VII. SIMULATION RESULTS

To obtain the designed bridge performance, the model seen in Figure 2 was used. This model indicates that the CAN/ATM bridge not only provides a service to interconnect a CAN system and an ATM LAN but also performs all the required functions for the communication between end systems. The performance of the designed bridge was evaluated from the utilisation of the CAN bus under different loads (Figure 4), total processing time of the bridge statistics (Figure 5) under different loads, and finally required buffer size for two ports (Figure 6).

From the figures 4,5 and 6, it is concluded that both sides of the bridge and both systems can support message traffic up to mean 60% utilisation of the CAN bus. 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 5, the bridge forwarding rate affects the process time of the messages only with the load more than 2000 messages per second in the ATM to CAN process. The total process time of the bridge is about 500 microseconds for both sides of bridge in the high remote message ratios (more than 50% bus utilisation). This value is acceptable for an interconnected CAN system. Another element affected by the load is the buffer capacity. As the loading of the buses increases, the

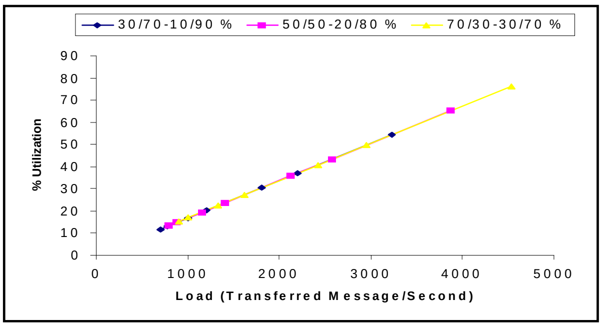

Figure 4: Utilization of CAN bus with different message ratios

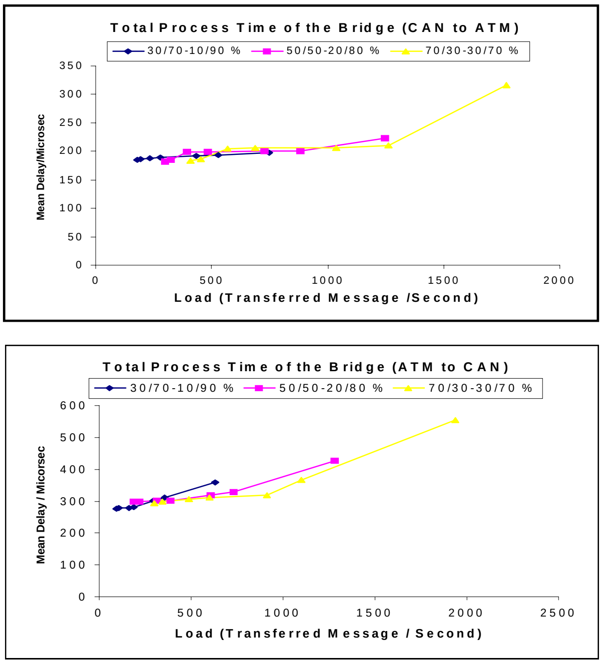

Figure 5: Total process time of the bridge

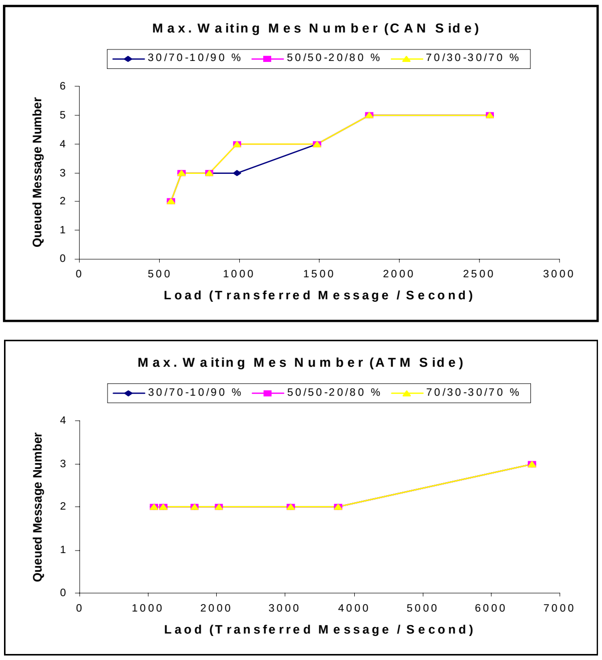

Figure 6:Queued message number (Required buffer capacity)

number of queued (buffered) message to be processed within the bridge increases. The result presented in Figure 6 quantifies the required buffer capacity, which is related to the number of queued messages, for various network loads. The number of queued messages is less than 10 on the both sides of the bridge. This results in each port of the bridge needing a memory which is capable of storing 10 messages of maximum length.

VIII. CONCLUSION

A bridge must provide a selective frame retransmission function and an interface operation, which allow communication between dissimilar systems. The objective of this research is to design a bridge that provides a service to achieve the interconnection of CAN and ATM networks. 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 ATM LANs. Secondly, the bridge decides whether or not to forward the frame. Thirdly, 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, utilisation of the bridge elements, and required buffer size which are related to the performance of the bridge are satisfactory in meeting the overall requirements.

References

1. Mclauglin, R.,, ‘CAN Controlling from cars to X-Rays’, IEE Networking, May 95, UK.
2. Ekiz, H., Kutlu, A., Powner, E.T.,: ‘Design and Implementation of a CAN / CAN Bridge’, Proc. of IEEE I-SPAN 96, June 1996, Beijing, China, pp. 507-513.
3. Ekiz, H., Kutlu, A., Powner, ET., : ‘Design and Implementation of a CAN / Ethernet Bridge’, Proc. of 3rd International CAN Conference, Paris, France, 1-2 October 1996, pp. 1117-1126.
4. Hawe, B., Kirby, A., ‘Transparent Interconnection of Local Area Network with bridges’, Telecommunication Network, vol. 3, no. 2, pp. 116-130, 1984.
5. Halsall F, Data communication, computer networks, open systems, Addison-Wesley Pub. 1996, USA.
6. BCAN Network Evaluation System with 80C552/ 80C196 CPU User Manual, I&Me, 1990, Germany.
7. Cisco Systems’ ATM Interface Processor, Product Announcements and Data Sheets, Cisco Company Home Page (www.cisco.com).