Performance Analysis of a CAN / CAN Bridge

H.Ekiz, A.Kutlu, M.D. Baba, E.T.Powner
School of Engineering
University of Sussex
Brighton BN1 9QT
E. Mail: h.ekiz@sussex.ac.uk

Abstract

The Controller Area Network (CAN) is a serial bus with high speed, high reliability, and low cost for distributed real time control applications. The CAN is a desirable, cheap solution for networks in industrial environments [1], but there is limit on the maximum length of a single CAN bus. A solution is to divide the CAN network into segments and connect them using bridges. Bridges are high performance devices that are used to interconnect LANs at the Logical Link Control (LLC) or Medium Access Control (MAC) level in the protocol hierarchy [2].

Unlike many serial communication protocols the CAN message contains no information relating to the destination and source addresses. Instead the message contains an identifier which indicates the type of information contained in the message. Because of this feature of the CAN message it is not possible to use traditional address based bridges to connect CAN segments.

The aim of this study is not only to design and implement a bridge to connect CAN segments based on CAN protocol features but also to do performance analysis of the designed bridge. The designed bridge is a transparent bridge, and uses a learning process to built-up a forwarding database.

1. Introduction

The CAN is a high performance and high reliable advanced serial communication protocol which efficiently supports distributed real-time control (in the field of automotive and industrial control) with a very high level of security [3]. The fast growing use of the CAN in many industrial applications soon resulted in a specific CAN-based network. In CAN-based distributed control systems, one of the potential problems is the size of distributed area. With a CAN bus, the physical length limitation of the bus is 2 Km at a data rate of 20 Kbit/s, but it can be maximum of 50m at 1Mbit/sec and 100m at 500Kbit/sec. In this case, the appropriate solution is to divide the network into segments and connect them using a MAC bridge [4]. With this approach, it is necessary to consider two subjects in some detail: both the CAN and the MAC bridge.

1.1. Controller Area Network Protocol

The Controller Area Network communication protocol is a contention-based serial communication. As an access method, the CAN uses CSMA/CR, Carrier Sense Multiple Access with Collision Resolution. The CAN although serial in nature is unlike many serial communication protocols; it contains no information relating to the destination or source addresses. Instead, the message contains an identifier which indicates the type of information contained in the message. The identifier is not only used to identify the message but also used in the arbitration mechanism. The CAN associates a priority with each message to be sent and uses a special arbitration mechanism to ensure that the highest priority message is the one transmitted. Broadcast to all nodes is inherent in this system and therefore data will be consistent in that either none or all of the nodes will receive the message. In other words, any node can have access to the bus for transmission of messages or any number of the nodes can receive and simultaneously act upon the same message [3].

In CAN, data is transmitted as a message consisting of between 1 and 8 bytes (octets). A message may be transmitted periodically, sporadically, or on-demand and is sent as a frame. The CAN device supports four different frame types: a. Data frame, b. Remote frame, c. Error frame, d. Overload frame.

CAN Data Frame: The data frame contains information to synchronise, to identify, to control and to save the data flow in addition to the user data. The format of the transmission of a data frame consists of seven fields as shown in Figure 1.

0-8186-7453-9/96 $05.00

The start of frame indicates the beginning of a message with a single dominant bit and synchronises all stations. This is followed by the arbitration field which contains an eleven-bit message identifier (ID) plus a remote-transmission-request (RTR.) bit. The RTR bit is used when a node wishes to request information from another node. It is always dominant in a data frame. When it is recessive ('1'), the frames becomes a 'remote' frame. Whilst the arbitration field being transmitted, the transmitter accompanies the transmission of each bit with a check to ensure that no higher priority message is being transmitted [5].

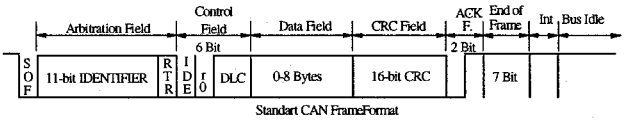


Figure 1. Standard CAN data frame format

The Control field consists of six bits. The two leading bits are reserved and are transmitted as dominant bits while the next four bits contain the code indicating the number of data bytes in the data field (the data length code DLC). The length of data field varies in the range from 0 to 8 bytes based on the value of the data length code (values in binary). The byte at the lowest address of the data segment will be transmitted MSB first.

The CRC field (Cyclic Redundancy Check) contains the CRC checksum (15 bits) followed by the CRC delimiter (1 bit). The CRC field contains a check word for detecting possible transmission interference. Therefore, the CRC code includes the start bit, arbitration field, control field, data field and CRC field. The CRC field provides strong error detection capability and MSB is transmitted first. After a message is completed the CRC field is checked. If any error is dedected in the frame, the whole frame will be retransmitted.

The ACK (Acknowledgement) Field consists of two bits, the ACK-slot and ACK-delimiter which are sent with a 'recessive' level by the transmitter. The ACK field allows the transmitter to be informed that the message has been received and that no error has been detected. While any transmitting node is sending the ACK_SLOT bit as 1, any receiving node which has received a frame correctly will send a 0 to the bus at the same time. Because the transmitting node listens to the bus, it will find the change and know that at least one station has received the message completely and correctly. Each data frame is delimited by the end of frame bit sequence, marks the end of the message, which consist of seven 'recessive' bits. There is a minimum interframe space (IFS) required between data frames. More detail can be found in [3].

1.2. MAC Bridges

Bridges are high performance devices that used to interconnect LANs at the Logical Link Control (LLC) or Medium Access Control (MAC) level in the protocol hierarchy [6]. A bridge not only overcomes the limitations (maximum length of single bus and number of nodes connected together) but also improves the whole system's characteristics such as performance, security, and reliability.

The bridges can be classified as a transparent and source routing bridge [7]. Each bridge type use a different routing method. There are three routing algorithms to enable a frame to reach a destination: the spanning tree (if a system is not sophisticated, the self-learning process is applied), the source routing, and spanning tree-source routing. In a transparent bridge, all routing decisions are made exclusively by the bridge. With this technique, the bridge can identify which stations are active on the media and builds its own address tables to control routing between two LANs. Transparent bridges perform three basic functions [6]: Learning of station addresses, Frame forwarding, and Resolving possible loops in the topology by using the spanning tree algorithm.

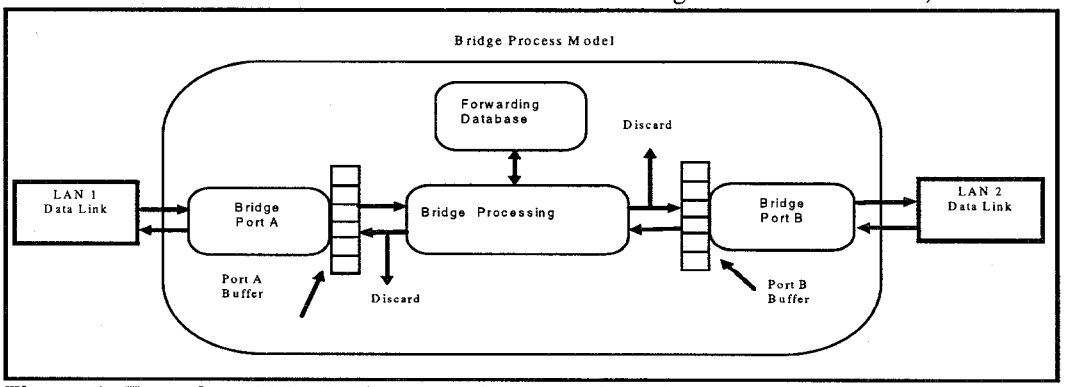


Figure 2. Functional block diagram of the bridge implementation

Figure 2 shows bridge functions in outline. 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 (discard) the frame (known as filtering). This decision to forward the frame to its intended destination is made by consulting the look-up table in the bridge, that contains the destination addresses in a form of database. 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 flooding process in the transparent bridge. This process is executed at the times when the bridge is first plugged in and at other predefined periods. When the bridge is first plugged in, the forwarding database (look-up table) is empty. The bridge needs to learn which addresses are local and which addresses are external. To build up the forwarding database which maps addresses, every incoming frame is transmitted to the other side of the bridge. This action is referred to as flooding [7]. As time goes on, the bridge learns the addresses of all stations interconnected via that bridge. This process is repeated in a defined period to notice whether there is any change in system or not.

The forwarding of data can take two forms: pass through and translation [8]. Pass through is the simplest way of forwarding. It is possible only when the incoming and outgoing frames from/to LANs have an identical frame format. In this case, the bridge forwards frames unchanged. Using this approach, a bridge provides a connection between two LANs that employ the same data link protocol. Translation is necessary when the LANs have different frame formats [9]. Therefore a translating bridge provides a connection capability between two LANs that employ different protocols at the data link layer.

2. Designing a CAN / CAN Bridge

The CAN / CAN bridge will be a pass-through bridge (both sides of bridge use same protocol) and it will use the learning process for routing. However, the learning process should be adapted for CAN features during the design.

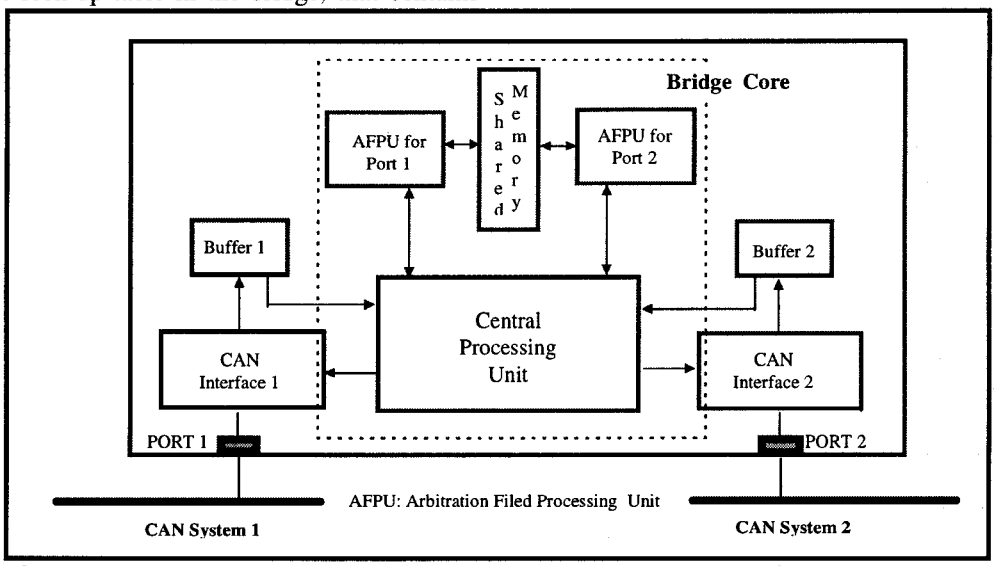


Figure 3. General block diagram of designed CAN to CAN Bridge

The bridge architecture can be modelled as shown in Figure 3. Assuming a two port bridge, the bridge consists of two network interfaces for CAN networks, two dedicated Arbitration Field Processing Unit (AFPU), a shared memory unit for the database (look-up table), and a central processing unit to provide necessary control functions.

The Central Processing Unit (CPU) should exhibit high performance to interpret frame relaying decisions from the AFPU and reschedule the received frames. The CAN Interface (CI) unit provides an interface between the bridge and the CAN systems. This unit performs not only CAN frame reception and transmission but also all other CAN protocol functions such as CRC processing. CI unit comprise two integrated circuits; the first is a microcontroller which performs the functions involved and the second is a CAN chip that implements all of the CAN protocols.

The AFPU is the heart of the bridge and it is responsible for generating frame relaying decisions. To achieve routing decision, the AFPU performs three processes: First, it extracts arbitration field from each received CAN frame. Second, it compares the contents of the extracted arbitration field with the contents of an arbitration fields in the database table. And third, it adds the arbitration fields to the list in defined periods (during flooding process) or discard it.

A major issue with the transparent bridges is the creation of the forwarding database [7]. To create and maintain the forwarding database, the flooding process is applied. At present, to create a forwarding database, possible criteria are destination addresses, bridge numbers and LAN numbers. But in the CAN system, none of these criteria can be used. Since it has no information about any of these criteria, consideration has to be given to a new criteria which can be obtained from the CAN frame. The
arbitration field seems the best field which can be used to create a forwarding database.

a frame is received by one of the CAN interface without an error and it is written in the involved buffer. If the central processing unit is not busy, it starts to execute the frame relay process to establish the routing information. The arbitration fields, one is extracted from the received frame and the other is read from the look-up table, are compared. If they match, the received frame is transferred to the CAN interface and is sent to the other CAN system. The comparison process continues until the last address has been read. Second, if the arbitration fields do not match, the time is checked whether it is flooding process time or not. If it is flooding process time, the frame is sent

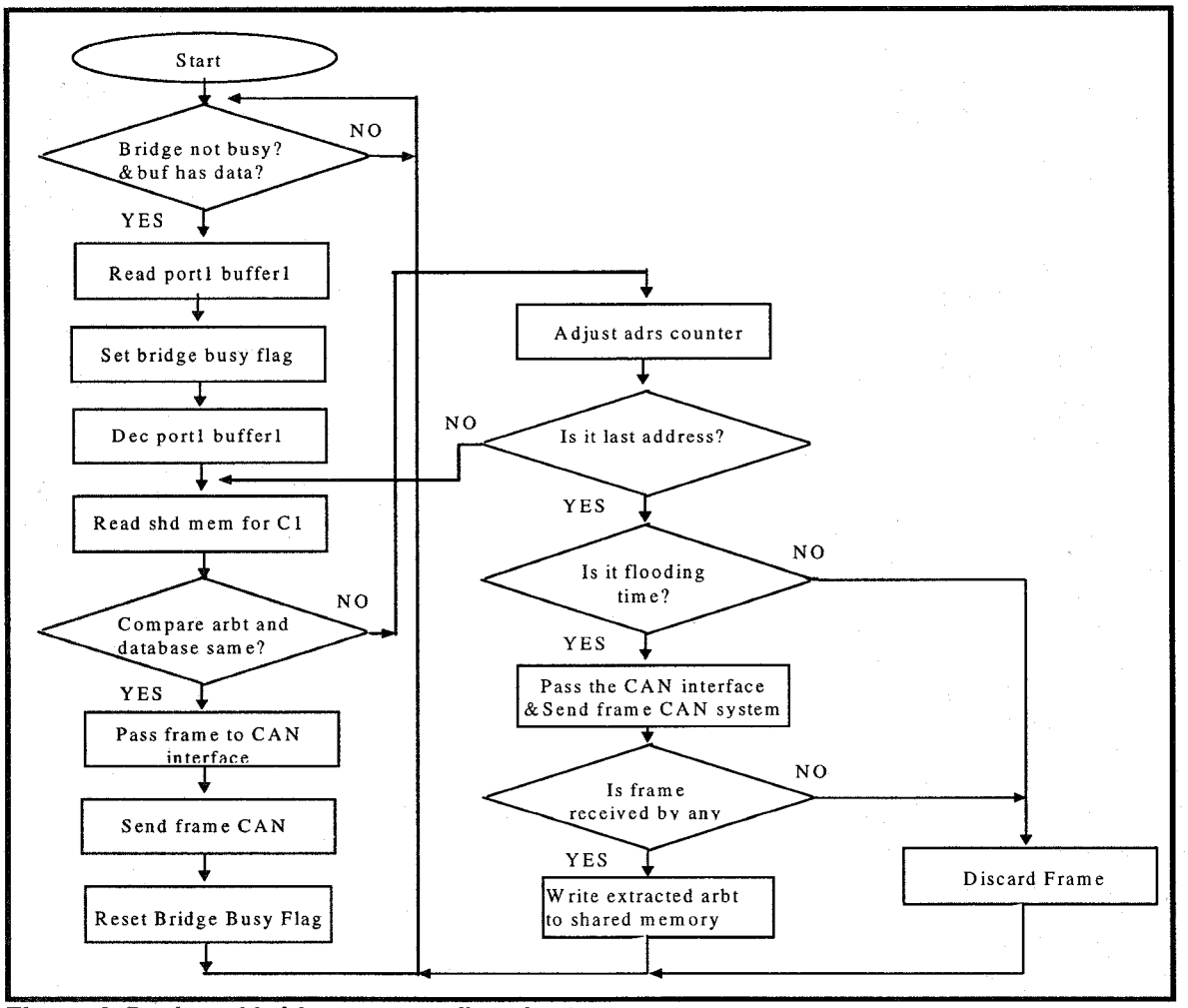


Figure 4. Designed bridge process flowchart

2.1. A CAN / CAN Bridge Process

In order to minimise the bridge delay in the bridge architecture, bridge elements should perform their processes in a reasonable time. This also helps the bridge to minimise the buffer size. The whole process performed in the designed bridge and its order can be summarised with a flowchart as shown in Figure 4. First, in the bridge,

to the other CAN system. Third, the look-up table is modified based on second bit of the ACK field. In the CAN protocol, when a node is transmitting a frame, it monitors that if any node is receiving the frame (by the ACK field).

If a node receives the frame, it changes the second bit of the ACK field from '1' to '0'. If this changing process occurs, the CAN interface sends a message to the bridge core to indicate that the arbitration field should be written

into the look-up table. However, if no change occurs, in the ACK field, the frame is discarded. Last, the received frame is either discarded or it is passed to the CAN interface to transmit across the other CAN system and the bridge core is set to read a new frame.

In the CAN protocol, if a transmitting node does not monitor that the ACK field has been changed, it assumes that the incoming frame had not been received successfully and retransmits the frame. In the bridged CAN system, this feature creates a problem during the learning process. This problem should be solved in order to create a database table. The solution is to set the SIE bit of the control register (Status change interrupt enable) to '1' and to check and change the LEC (Last Error Code) bits of the Status Register.

If SIE is enabled (set to '1'), an interrupt is generated in two cases: when a CAN bus error is deduced or a transfer

is completed successfully [10]. The interrupts which occur due to successfully completed transfers can be ignored, because these interrupts do not cause any problem. The important interrupt (interrupt message) is the one which occurs due to changing of the LEC bits in the status register (these bits are changed based on type of error). This interrupt message can be used to stop the retransmission of the frame in the bridged system. With this approach, when the CPU recognises any interrupt from the status register, it stops the retransmission and resets the LAC 0-2 bits of the status register ('0' means no error). When CAN transceiver read '0' in the status register, it assumes that the transmitting process has been completed. More information about the registers of CAN microcontroller can be found in [10].

2.2. Construction of a Data Base Table

In the bridge architecture, two important factors that affect the bridge's ability to generate frame relaying decision are the organisation of the address list and the techniques used to perform the look-up table search algorithm. In the designed bridge architecture , it was assumed that the binary table search technique had been used. Well-known list search techniques used to perform the destination address search algorithm are explained in [11]. The binary table search technique needs an ordered list but has a fast comparison time. With the binary search algorithm, the value (arbitration field) is compared to the entry at the mid-point location within the list. Depend upon the relative magnitude of this list entry and arbitration filed, the next list entry to be examined will either be at the quarter or three quarter position in the list. Hence, each comparison either locates the arbitration field value or eliminates half of the remaining list entries from further consideration.

On the other side, when the bridge is in progress, location of nodes or process of any node can be changed.

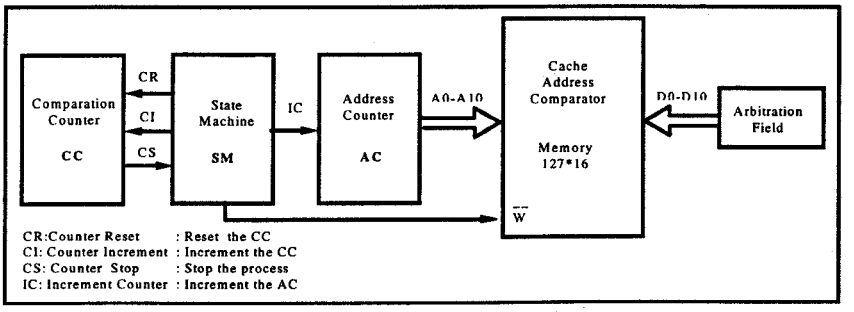


Figure 5. Block diagram of the comparison process mechanism

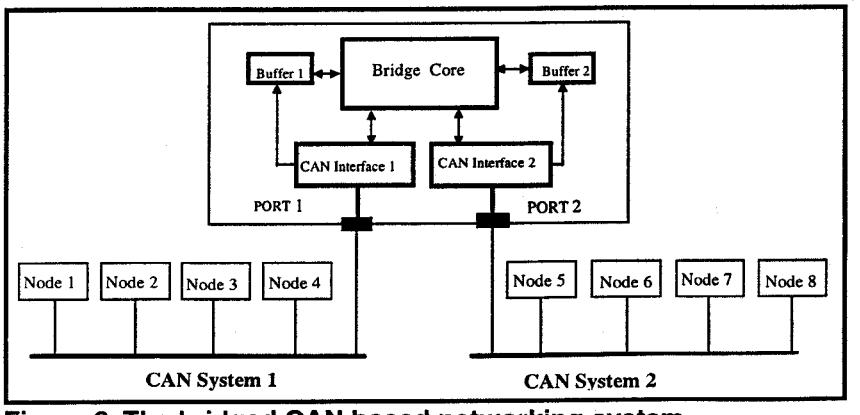


Figure 6. The bridged CAN based networking system

This change can be recognised by the bridge during the next flooding process. If it is assumed that a new forwarding database table is created in each flooding process time, it will take a long time, and the number of comparisons will be the same until the next flooding process. Instead, if a check the entries in the forwarding database is made using an inactivity timer which is associated with each entry in the database and reflects the last used time of each entry in the database, the database can easily be updated. Whenever a frame is received from a node, the corresponding timer for that entry is reset. If no frames are received from a node until the next flooding time, this entry is removed. This means that the new flooding process (except the first one) starts with a table which has been updated. In addition, if it is assumed that any entry whose corresponding timer has not been reset within the predefined time interval, the related data is removed from the list. This process limits the size of the database since it contains only those nodes that are currently active.

2.3. Comparison of Arbitration Fields

In order to extract the identifier information from the received frames, the start of the frame must be deduced. A CAN frame, as shown in Figure 1, commences with a start of frame (SOF) bit. The first single dominant bit ('0') indicates the beginning of the frame. After the recognition of SOF, the extraction of any field from the frame can be performed by using a shift register and a binary counter [11]. In the designed system, after the extraction of the arbitration field, the next step to be performed is to compare the extracted arbitration field with the content of the forwarding database (arbitration list).

The comparison process can be performed on a state machine which contains two counters and a cache address comparator (Figure 5). The detail of the comparison process can be found in [4]. The state machine (SM) provides involved commands and signals for other elements of the comparison process mechanism. At the beginning of the comparison process, the comparison counter (CC) is reset and incremented by 1 after each comparison. The address counter (AC) helps to read an arbitration field from the appropriate memory address. After a comparison, if compared values do not match, the AC is re-set to an appropriate address. A cache address comparator (e.g. TMS 2150) compares the value of the data bus with the content of the RAM location which is currently being addressed.

3. The Modelling Environments

A commercial simulation package that has been developed to model network systems and network devices was used to model the bridge shown in Figure 3. In the simulation, the simulation program models three processes, namely; bridge operation, bridge-CAN interconnection, and the transmission and reception of frames. It was assumed that a microprocessor, M68000, performs the bridge core processes. After defining each element, seen in Figure 3, the designed bridge was used to
interconnect two CAN systems as shown in Figure 7. For simplicity the simulation model is based upon the following assumptions.

a) The features of the CAN evaluation board from I&Me products were chosen in the implementation to define the CAN interfaces.

b) The process times were measured as delays: receive frame and write buffer time, bridge core process time, and frame transmit or discard time.

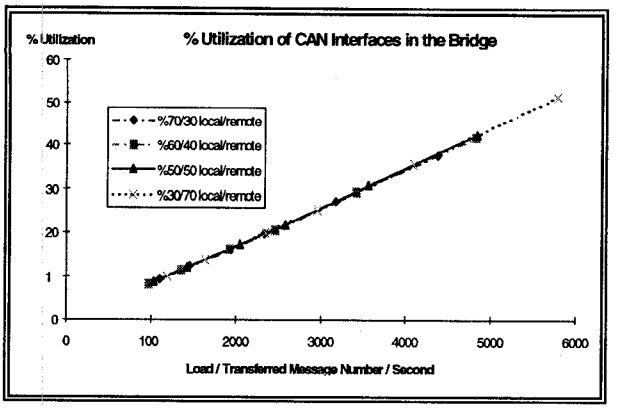
c) To define the features of the peripheral devices, M68000 peripheral devices features were chosen.

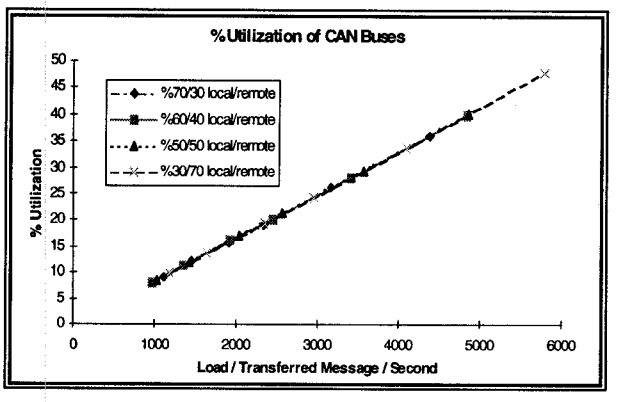
d) The CAN system bus speed was chosen as 1Mbit/sec.

e) In the networking system (Figure 6), the message traffic was defined with various local message/remote message ratio as % 70/30, 60/40, 50/50, 30/70.

4. Simulation Results

To obtain the performance of the designed bridge, the model seen in Figure 6 was used. The system was loaded with various local/remote message ratios. This model indicates that a CAN to CAN bridge overcomes the limitations which mentioned in previous sections (size of the CAN and the number of nodes connected together). The performance of the designed bridge was evaluated from the utilisation of the bridge and the bus systems (Figure 7) and total processing time of the bridge statistics (Figure 8) with different loads. The utilisation of the bridge elements and the CAN buses is less than %50 utilisation when the load is about 5000 frame/second. This indicates that the designed bridge performance is adequate to interconnect the CAN segments. The total process time of the bridge is acceptable for the CAN systems. Although when the remote message ratio is high, the process time increases, the process time that has almost twice frame transmit/receive time is acceptable. If we consider the two process delays together, the total process time and the PECAN (CAN Interface Units) waiting time for the retransmission (Figure 9), the whole process delay reaches to almost 3 frames. In addition, the CAN buses request delay to access the bus is quite small (Figure 9). As a result it can be said that the throughput of the designed bridge is satisfactory in the system.





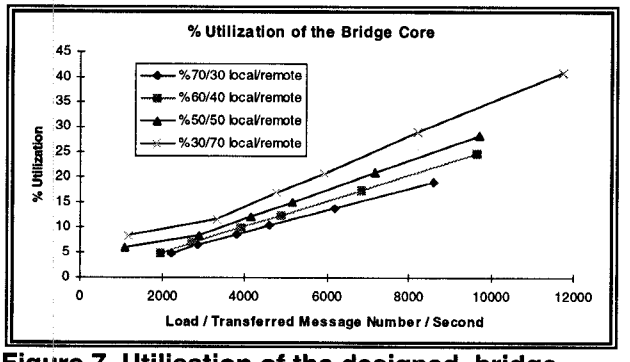


Figure 7. Utilisation of the designed bridge elements and CAN buses

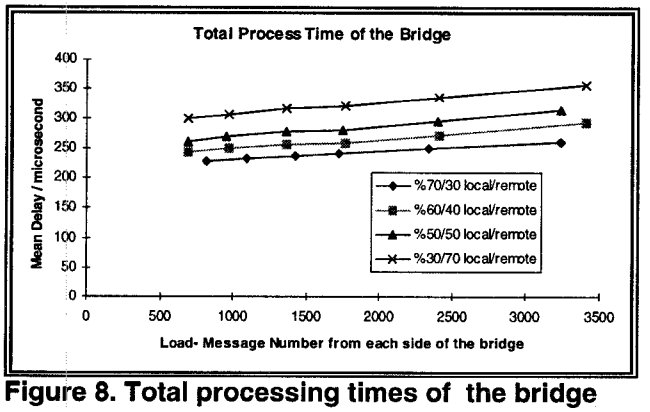
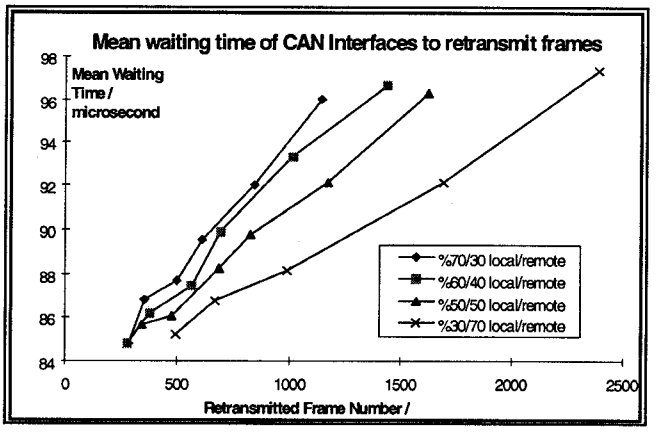


Figure 8. Total processing times of the bridge process (transmit or receive) length.

In the bridge design, one of the most important parameters is the required buffer size. Although the required buffer size changes depending upon the message traffic, the designed system gives a general idea about the required buffer size. The size of buffer, in the worst case, must be minimum 5*108 bits in the designed system (Figure 10). This means that there are 5 frames in the queue in the worst case.



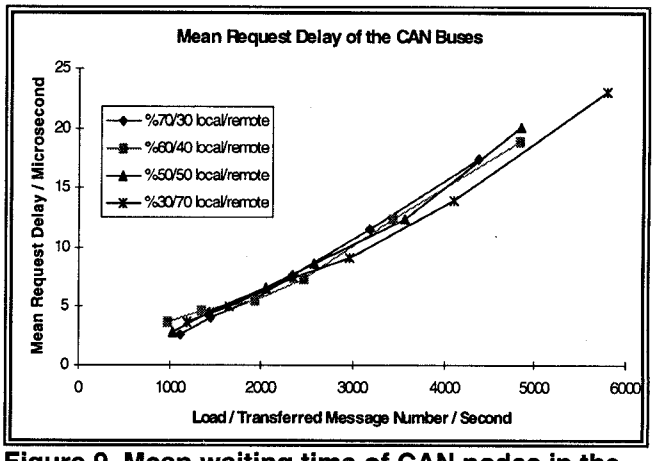


Figure 9. Mean waiting time of CAN nodes in the bridge for retransmission and request delay on CAN buses

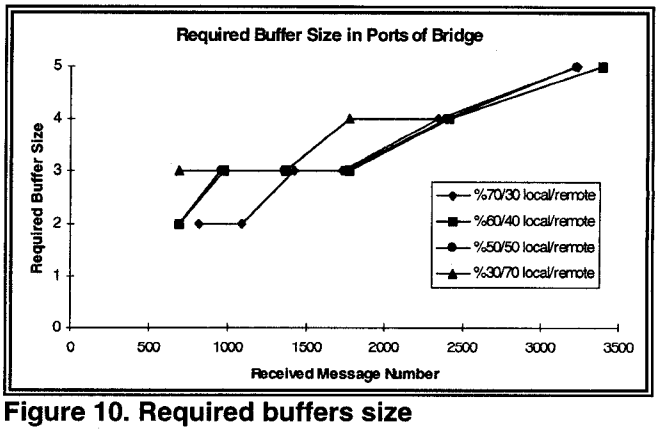


Figure 10. Required buffers size

5. Conclusion

A bridge must provide a selective frame retransmission function which allows communication between nodes located on the other side of the bridge. The objective of this research has been to design a bridge that provides a selective frame retransmission function in the interconnected CAN systems. The designed bridge has fulfilled the objectives and it overcomes the limitations mentioned in the previous sections. In summary, the operational model of the bridge includes three phases of operation. First, the bridge receives a frame from a CAN for forwarding. Second, the bridge decides whether or not forwarding the frame. Third, the bridge transmits the frame to the next CAN. The designed bridge is a transparent bridge and uses learning process to built-up the forwarding database. It uses the arbitration field of the CAN frame for forwarding decisions. The both parameters, delay and throughput, which are related to the performance of the bridge are satisfactory.

References

[1] Mclauglin R., *CAN Controlling from cars to X-rays*, IEE Networking May 95, UK.

[2] Fetterolf P.C., *Design of Data Networks with spanning tree bridges,* Proc. IEEE International Conference Systems, MAN and cybernetics, 1990, USA.

[3]. *Controller Area Network (CAN), LAN in vehicle communication protocol,* SAE JI583 Mar90, SEA Information Report, pg. 20. 226-248

[4]. Ekiz H., Powner, ET., Kutlu A. "*Design and Implementation of a CAN/CAN Bridge*", Proceedings of IEEE ISPAN'96 Conference, pg. 507-513, Beijing, China, 12-14 June 1996.

[5]. *Automotive Electrical systems/CAN,* Automotive Handbook, pg. 777-778, 3. Edition, Robert Botch GmbH

[6]. Accepted, Ekiz H., Powner, ET., Kutlu A. "Design and Implementation of a CAN / Ethernet Bridge", 3rd International CAN Conference, Paris, France, 1-2 October 1996

[7] Hallsall F, *Data communication, computer networks, open systems*, Addison-Wesley Pub. GM. 1992, USA.

[8] Have B., Kirby A., *Transparent Interconnection of Local Area Network with bridges*., Telecommunication Network vol3, no 2, pp 116-130, 1984.

[9] Tanenbaum A.S., *Computer Networks*, Prantice Hall, 1989 USA.

[10] 82527 Intel Serial Communications Controller Architectural Overview, October93

[11] Linge, N., *The interconnection of LANs using bridges*, Salford University, PhD Thesis, 1987.