RDM+: a New Mac Layer Real-Time Communication Protocol

Mojtaba Sabeghi
Computer Engineering Lab
Delft University of Technology
Delft, the Netherlands
sabeghi@ce.et.tudelft.nl

Mahmoud Naghibzadeh
Computer Engineering Dept
Ferdowsi University of Mashhad
Mashhad, Iran
naghib@um.ac.ir

Koen Bertels
Computer Engineering Lab
Delft University of Technology
Delft, the Netherlands
k.l.m.bertels@tudelft.nl

Abstract

Many real-time communication protocols have been studied to guarantee the communication requirements of distributed real-time systems. But, current techniques lack all or most of these requirements specially bounded message delivery time. In this paper we have proposed a Mac layer protocol called RDM+ with concepts similar to Round Data Mailer multi layer protocol [1]. Our simulation results show that this new protocol has the potential to be used in today's automotive systems.

1. Introduction

The common feature of modern distributed systems are a large number of devices interconnected together to perform the desired operations. Examples of these systems include intelligent vehicle systems. At one time, a radio was likely the only electronic device in an automobile, but now almost every component of the vehicle has some electronic feature such as the Engine Control Module (ECM), the Timing Control Module (TCM), the Anti-lock Braking System (ABS) and body control modules (BCM) which we call them Electronic Control Units (ECUs) [2].

An electronic control module typically gets its input from sensors that it uses in its computation. The modules need to exchange data among themselves during the normal operation of the vehicle. The vehicle network is the medium of data exchange. Needless to say, that these communications must be done with a real-time protocol.

To achieve real-time communication, two things are necessary; real-time protocols and real-time networks. A real-time network is very simply one that displays timely and reliable behaviors.

These networks may be data network or control network depending on the information exchanged. Generally speaking, data networks use large data packets and relatively infrequent transmission and most of the times these data are not time critical data. On the other hand, control networks must shuttle small but frequent packets among a relatively large set of nodes to meet the time critical requirements.

In order to guarantee that the timing requirements of all the tasks are met, the total communications delay between sending and receiving a message must be bounded. This total communications delay is often termed the *end to end* delay, the time between a message being queued by the sending task and the message fully arriving at the receiving task. This delay is composed of the *access delay*, the *propagation delay*, and the *delivery delay* [3].

The access delay is the time a message queued at the sending processor spends waiting for the use of the communications media. With shared media network architecture, a processor must compete with other processors for use of the media [4].

The propagation delay is usually defined as the time taken for the data to reach the destination processor once physically sent by the source processor. In this paper we are concerned with packet networks, and consider propagation delay to mean the time taken between a packet starting to be transmitted and the entire packet finally arriving at the destination processor. The delivery delay is the amount of time taken to process the incoming data and deliver it to destination tasks.

There are several network types and protocols used in vehicles by various manufactures to address aforementioned requirements. Some of the more common technologies interconnecting ECUs today are the Controller Area Network (CAN) [5, 6], Time Division Multiple Access (TDMA), and Carrier Sense Multiple Access (CDMA) [7].

2. RDM+ Protocol

This protocol is a token based protocol. The idea of using a token comes from the timed-token MAC protocol. A large amount of research regarding the token protocols

has been done and still almost all real-time network protocols use some sort of token algorithm to provide QoS guarantees. The main advantage of using a token is collision prevention. It means that only one node can transmit messages across the network at any given time, because it is the only one holding the token. Secondly, by letting each node hold the token for a pre-designated time and using a deterministic algorithm to select the next node that is eligible for the token, QoS guarantees can be given [8].

In RDM+ protocol, a logical circular ordering of nodes is assumed and a critical message moves clockwise or counterclockwise around the circle, from one node to its neighbor. The critical message is a message that carries real-time data/results between nodes. Only one such a message exists for the whole system. This message acts exactly like a postman who circulates around the city and delivers postal packages. This critical message is called token which is a little bit different from the so-called tokens in the other protocols. In fact this token is a carrier of the data which have to be sent [13].

Although RDM+ is not necessarily a ring based protocol and just a logical circular ordering is enough for it but ring based implementation is suggested. Here, we summarize some ring benefits which often make it more useful in network systems:

1. Data is quickly transferred without a bottle neck.
2. The transmission of data is relatively simple as packets travel in one direction only.
3. Adding additional nodes has very little impact on bandwidth

The main contribution of this research is that we have considered RDM+ as a protocol designed for MAC layer rather than a multi layer protocol. The MAC protocols are in charge of managing the sharing of the transmission media. Moreover, the aim of a MAC protocol is to control the interference and competition among users while optimizing overall system performance. Besides, it is the critical algorithm for determining the ability of a network technology to support, in an efficient and fair way, both real-time and non-real-time traffic. All the aforesaid are among the responsibilities of a real-time communication protocol.

3. Frame Structure

RDM+ has eight types of messages which have been presented in Figure 1. The Bits column contains the bit series that distinguish between different types of messages.

Message Type I is the carrier of real-time data which circulates around the ring and delivers the data. After receiving the critical message, each node must send a Message type II, acknowledgement, to the sender which is its neighbor. If the receiver does not receive the message accurately, it will ask for the message to be sent again by message type III. Messages type IV and V can be used in case of node failures. When the system is started, the coordinator will introduce all real-time data to all nodes. It will release a message type VI in the ring. Messages type VII and VIII are non-real-time message types. A non-real-time message of type VII can only be sent when there is a real-time break. The coordinator announces real-time breaks when the circulation time of the critical message is too low. By too low we mean, there is enough time left to be able to send at lease one non-real time message.

Type	Message Action	Bits
I	Critical Message	000
II	Acknowledgement	001
III	Request to resend	010
IV	Are you alive?	011
V	I'm alive	100
VI	Reconstructuring	101
VII	Non-real-time Message	110
VIII	Real-time Break	111

Fig.1. RDM+ messages

The frame structure for each type of the messages has been illustrated in figure 2. Here, there is a brief explanation of the frame fields.

- Starting Delimiter — consists of a special bit pattern denoting the beginning of the frame. The bits from most significant to least significant are J, K, 0, J, K, 0, 0, and 0. J and K are code violations. Since Manchester encoding is self clocking, and has a transition for every encoded bit 0 or 1, the J and K coding violate this, and will be detected by the hardware. This field exists in all types of messages.
- Access Control — this is a 12 bits filed. The first 3 bits determine the message type. The next bit will be set when the message cross out the coordinator. The last 8 bits contain the coordinator id. This field exists in all types of messages.
- CRC — a field used to store the calculation of a CRC for frame integrity verification by the receiver. This field exists in all types of messages.

- Ending Delimiter — the counterpart to the starting delimiter, this field marks the end of the frame and consists of the following bits from most significant to least significant: J, K, 1, J, K, 1, I, E. I is the intermediate frame bit and E is the error bit. This field exists in all types of messages.
- Load ID — each datum in the system must have a unique ID with which it will be distinguished from the other data. This field exists in message type I and message type VI.
- Load Offset — a field which determines the offset of an specific datum in the critical message. It will simplify the message analysis in each node. This filed exist just in message type VI.
- Load — a variable length field of 0 or more bytes, the maximum allowable size depends on the ring speed containing MAC management data or upper layer information. This filed exist in messages type I and VII. In fact, these two messages are those which carry data across the network.
- Length — a field which specifies the frame length.
- Destination — used to specify the destination's ID. Each node in the system must have a unique id with which it will be addressed.
- Source — contains the sender's ID.

Start Delimiter	Access Control	Length	Load1 ID	Load1	…	CRC	End Delimiter

(a) Message type I

Start Delimiter	Access Control	End Delimiter

(b) Message type II, III, IV, V

Start Delimiter	Access Control	Length	Load1 ID	Load1 Offset	…	CRC	End Delimiter

(c) Message type VI

Start Delimiter	Access Control	Length	Destination	Source	Load	CRC	End Delimiter

(d) Message type VII

Start Delimiter	Access Control	Time	End Delimiter

(e) Message type VIII

Fig.2. Frame structures

4. Performance Evaluation

The performance metrics of network systems that impact the requirement of control systems are access delay, transmission time, response time, message delay, message collision, packet size, network utilization, deadline meeting, etc. In real-time control networks two main criteria are: bounded time delay and guaranteed transmission [12].

The End-to-End Delivery (EED) time of a message depends on these factors: (1) the amount time that is passed, since the message is generated, until the node that generated the message is granted to use the

communication media and the message's turn has come to be transferred, R_d, (2) The processing delay before sending the message to the hardware media, P_d, (3) The initial delay for the first bit to be received by the receiver, F_d, (4) The speed of the media in bits per second, S, and (5) The message length in bytes, L. The end to end delay can be easy formulated for RDM+. Simply, if node *i* sends the critical message to node (i+1mod *n*), then the critical message delivery time, CMD, is:

$$CMD_{i,\, i+1 \bmod n} = R_{di} + P_{di} + F_{di} + 8 * S * L \qquad (1)$$

In this research a thorough simulation has been done and the protocols have been compared base on the following metrics;

- Detail end to end delay analysis for both real-time and non-real-time traffics [9].
- Network access time which is the maximum time a node must wait before transmitting a packet into the network [10]
- Protocol Overhead to the network bandwidth or bandwidth utilization which can be discussed in from of the fraction of control messaged (ACK …) and then, we can estimate the real-time capacity of the network. (How much real-time data can be send over the network) [11]
- Message Delivery: Messages can be lost either due to deadline expiry, or station crash. Even in a fault-free system message loss due to deadline expiry cannot be prevented [9].

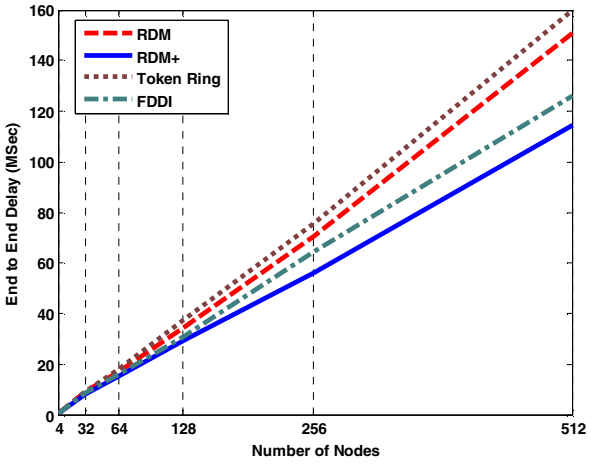

Fig.3. End to end delay

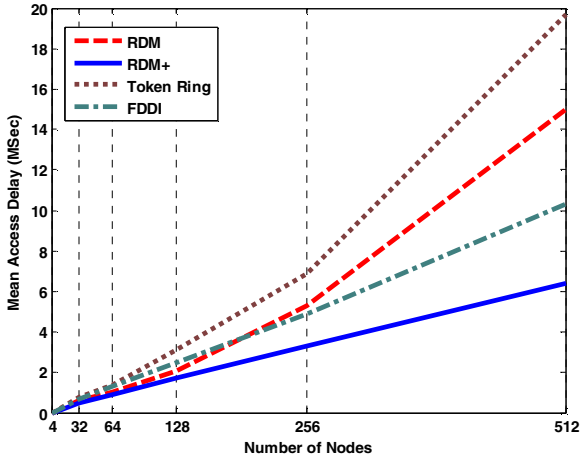

Fig.4. Mean access delay

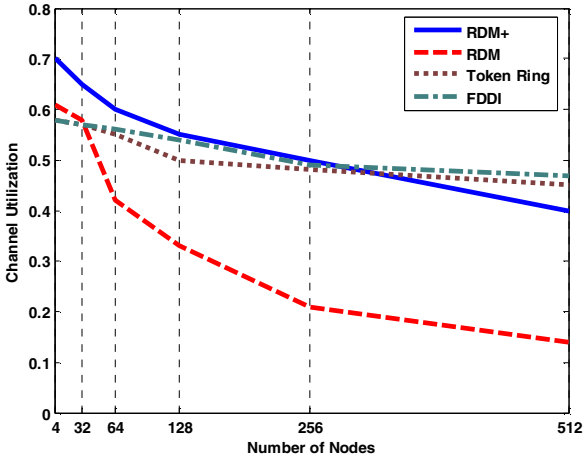

Fig.5. Channel utilization

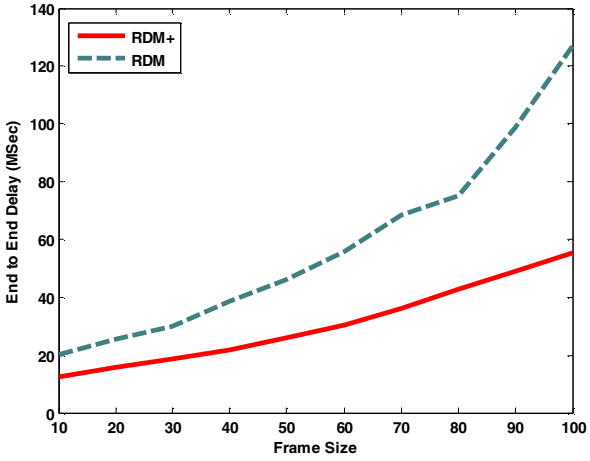

Fig.6. End to end delay with fixed number of nodes

Figure 3 represents the end to end delay diagram comparing the RDM, RDM+, Token ring and FDDI. Figures 4 and 5 show the other metrics' results. These metrics are mean access delay which is the maximum time a node must wait before transmitting a packet into the network and channel utilization. Furthermore, in figure 6, we have compared just RDM and RDM+, with constant node number and varying packet size. The metric with which the protocols have been compared is end to end delay.

5. Summary

In this paper we addressed the RDM+ protocol which has the potential to be used in today's automotive systems. Our Comparisons shows that RDM+ performs better for all performance metrics discussed. Furthermore, a comprehensive formal analysis of the protocol is underway.

6. References

[1] H. Abachi and M. Naghibzadeh, A message-passing protocol for small-scale distributed real-time systems, *World Automation Congress*, 2004

[2] T. Nolte, H. Hansson, L. Bello, Automotive Communications - Past, Current and Future, *10th IEEE International Conference on Emerging Technologies and Factory Automation*, 2005

[3] F. Lian, J. Moyne, D. Tilbury, Performance evaluation of control networks for manufacturing systems, *ASME-Dynamics Systems and Control Division*, 1999

[4] G. Agrawal, B. Chen, W. Zhao, S. Davari, Guaranteeing synchronous message deadlines with the timed token medium access control protocol. *IEEE Transactions on Computers*, 43(3), Mar. 1994.

[5] L. Almeida, J.A.G. Fonseca, P. Fonseca, A flexible time-triggered communication system based on the Controller Area Network: Experimental results. *In Proceedings of the Fieldbus Conference*, September, 999.

[6] L. Almeida, P. Pedreiras, J.A.G. Fonseca, The FTT-CAN protocol: Why and how. *IEEE Transactions on Industrial Electronics*, 49(6), December, 2002.

[7] M. Alves, E. Tovar, and F. Vasques. Ethernet goes real-time: a survey on research and technological developments. *Technical Report HURRAY-TR-0001*, Polytechnic Institute of Porto — School of Engineering (ISEP-IPP), January, 2000.

[8] F. Buzluca and E. Harmancı. Dynamic synchronous bandwidth allocation scheme for hard real-time communication in FDDI networks. *IEE Proceedings Computers and Digital Techniques*, 148(1):15–21, January, 2001.

[9] J. Chen, Z. Wang, and Y. Sun. Real-time capability analysis for switch industrial Ethernet traffic priority-based. *In Proceedings of the 2002 IEEE International Conference on Control Applications*, pages 525–529, September, 2002.

[10] Y. Choo and C. Kim. Guaranteeing periodic communication services in a multiple access network using token with timer. *Technical Report TP01ISA1109*, The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC, USA, 2001.

[11] H. Hoang and M. Jonsson. Switched real-time Ethernet in industrial applications – asymmetric deadline partitioning scheme. I*n Proceedings 2nd International Workshop on Real-Time LANs in the Internet Age*, July, 2003.

[12] M. Sabeghi, M. Naghibzadeh, Performance Assessment of a Distributed Real-Time Control System Utilizing RDM and RDM+ Protocols for Communication. *2ed CoNext Conference*, December, 2006.

[13] M. Naghibzadeh, Round Data Mailer Message, *IEEE International Conference on Systems, Man and Cybernetics*, October. 2002