2007 IEEE International Conference on Signal Processing and Communications (ICSPC 2007), 24-27 November 2007, Dubai, United Arab Emirates

Datalife Time Analysis in RDM+ Real-Time Communication Protocol

Shabnam Mirshokraie[1], Mojtaba Sabeghi[2], Koen Bertels[2], Mahmoud Naghibzadeh[1]

[1]Computer Engineering Department
Ferdowsi University of Mashhad
Mashhad, Iran
{mirshokraie, naghibzadeh}@um.ac.ir

[2]Computer Engineering Laboratory
Delft University of Technology
Delft, the Netherlands
{sabeghi, k.l.m.bertels}@ce.et.tudelft.nl

ABSTRACT

Real-time systems are being used increasingly in control applications such as in automobiles, aircraft and process control. Real-time communication protocols are designed in order to satisfy basic requirements of these systems such as reliability, safety and in time message delivery. One of the most important requirements of such critical systems is in time update of real-time data which has to be facilitated by these protocols. CAN (Controller Area Network) [1] is one of the important and common protocols which is widely used in such an environment. RDM (Round Data Mailer) [2] and RDM+ [3] are two new protocols that have been designed to fulfill the requirements of distributed real-time systems. In this paper, we are trying to compare RDM+ and CAN protocols based on data lifetime which is the time interval between two consecutive updates of a real-time datum. The obtained results show that RDM+ performs better and provides fresher data to the consumers.

Index Terms**—** RDM+, Datalife time analysis, Real-time communication protocol

1. INTRODUCTION

System safety, reliability, and timeliness are the most important aspects of a real-time system. These systems have been defined in the literature as: "systems in which the correctness of the computations not only depends upon the logical correctness of the computation but also upon the time in which the result is produced. If the timing constraints are not met, system failure is said to have occurred." [4, 5]

In a distributed real-time system, some nodes such as sensors produce data and some other nodes consume those data in their computation. The produced data should be sent to the consumer in a distributed real-time system via a real-time communication protocol. Usually, the consumer stores the received data in the local memory to use it as it is required. The problem is that this data is valid just for a certain amount of time and should be updated before that time finishes.

The most important performance metrics of the real-time protocols that impact the update rate of real-time data are access delay (the time spent from when a MAC frame reaches the head of a MAC layer queue to the time when this frame is successful transmitted), transmission time (the time length it takes a bit to go from the start of the link to its destination node), message delay (the difference between the time when the source node submits a message to be sent and the time when the destination node receives this message), message collisions (percentage of collision), message throughput (percentage of packets discarded), packet size, network utilization (ratio of the total time used to transmit data and the total running time) and determinism boundaries [6, 7].

TDMA (Time Division Multiple Access), CAN (Controller Area Network), Token Ring/IEEE 802.5 and FDDI (Fiber Distributed Data Interface) are the most important protocols that are presented in order to satisfy these requirements [8, 9, 10]. But there are some problems with each of these protocols.

In TDMA, the time slice is reserved for every node regardless of whether it has any messages to be sent or not. The length of the time slice assigned is based on the worst case message passing time not the actual requirements. This influences the overall system performance which may dramatically decrease it [2, 11].

CAN has a wide range of usages in control systems but its major disadvantages are low throughput and the slow data transfer rate. It is also not suitable for transmission of large data sizes [2, 6].

Token Ring has some other difficulties. In order to find the next highest priority message a token must travel at least one complete cycle through the ring and as a result awaiting messages may not be delivered in time [6, 11, 12].

Finally, FDDI provides some useful concepts, but it considered to be too expensive because of the high cost of its interfaces which is 5-10 times more than Ethernet and 3-5 times more than Token Ring. Also, FDDI concentrators are expensive [11, 13, 14]. Because of these problems, in aforementioned protocols it is not possible to make sure that every real-time data is updated in time [12].

Recently, a new protocol called RDM+ (Round Data Mailer) [3] has been proposed which utilizes the RDM message passing mechanism [2, 12]. RDM message passing mechanism tries to be conformable with real-time systems requirements. RDM+ protocol employs the RDM message structure to overcome many problems like

1-4244-1236-6/07/$25.00 © 2007 IEEE

token traveling and inefficient use of bandwidth. It can also guarantee the in time update of each real-time data.

In the original version, the RDM is multilayer protocol [12] in contrast with RDM+ which is a MAC layer protocol. RDM+ with its different and improved structure shows better performance than RDM [3, 15]. It has been designed for MAC layer to exploit modularity in the protocol design and to be used in conjunction with any protocols which are in the market for other layers of the multilayered communication protocols [3, 15].

The rest of the paper is organized as follows. Section 2 presents an overview of RDM message passing mechanism. The comparison metrics and the obtained results have been covered in sections 3 and 4 following by the conclusion section.

2. RDM MESSAGE-PASSING MECHANISM

To utilize RDM message structure and RDM message-passing mechanism, a logical ring ordering is assumed in order to connect the nodes together.

RDM message passing mechanism proposes eight types of messages to manage all the messaging requirements of a distributed real-time system. These eight types are (I) critical message, (II) acknowledgement in reply to receiving the critical message, (III) request to resend the last message when the critical message is not received properly, (IV) "Are you alive?" message which is sent when the destination node does not send an acknowledgement within a predefined period, (V) "I'm alive" message, (VI) restructuring message which is broadcasted by the coordinator in the system initialization or re-initialization phase to introduce the message type I structure, (VII) non-real-time message and (VIII) real-time break message.

In RDM message passing mechanism, the most important message is message of type I or the critical message. There is only one such message that travels between nodes. Critical message circulates clockwise (or counterclockwise) around the logical ring and carries all the real-time data/results. A detailed presentation of this message and its structure can be found on [3, 15]. When a given node receives the critical message, it has the permission to load its real-time data on the message. Then, the node will send the message to its clockwise (counterclockwise) neighbor.

A detailed message delivery analysis and fault tolerance aspects of this messaging mechanism has been studied in [16].

3. COMPARISON METRICS

A key characteristic of the real-time data is that they are valid for the consumer for a pre-defined amount of time and the data which are older than this, are not useful. Therefore, these data should be updated before expiration of this pre-defined time. We define *data validity time* as the allowed time duration after the last update of real-time data to be usable by the consumer.

In our simulation, two main performance factors are computed: *average datalife time* and *invalid data access ratio*. In order to do this, first we specify life time and also validity time for each real-time data. Datalife time is the time interval between two consecutive updates of the real-time data. In this simulation we compute average of this quantity for all the consumers (consumers are those nodes that use data which are produced by other nodes in the system). Datalife time has an inverse relationship to the update rate of real-time data, meaning the faster the update rate, the shorter the datalife time of a given piece of data.

If the life time of a given real-time data exceed its validity time, this data is assumed useless and it is called invalid data. As a result, if a node gets access to this invalid data then it is called invalid data access. Thus, invalid data access ratio is defined by the ratio of the number of invalid data accesses to the total number of data accesses (in this paper invalid data access ratio is computed in percentage form).

In the next section, we will study and compare RDM+ and CAN protocols based on these metrics. The comparisons have been done for both RDM+ and CAN in two different cases: messages with size of maximum 8 bytes and messages with size of maximum 1000 bytes. These cases have been considered to study the protocols performance for small data size as well as large data size.

Each node is the consumer of maximum five real-time data. Data validity time is assumed to be less than 1500 milliseconds in case of messages with size of maximum 1000 bytes, but it is assumed to be less than 500 milliseconds in case of messages with size of maximum 8 bytes. It should be mentioned that all of these numbers are randomly generated.

4. SIMULATION RESULTS

The average datalife time and invalid data access ratio for messages with size of maximum 1000 bytes are shown in figure 1 and 2. It should be mentioned that the curve fitting technique is used in drawing the average datalife time and invalid data access ratio diagrams. As we can see in figure 1 and 2, RDM+ performs better than CAN with a considerable difference.

There are two main reasons for different behavior of RDM+ and CAN, in this case. The first and more important reason is that RDM+ utilizes the RDM message-passing mechanism. As we mentioned before, in RDM message-passing mechanism a collection of all real-time data encapsulate in one so-called critical message. In fact, there is only one message that circulates around the logical ring and has the responsibility to update each real-time data which is needed by other nodes. This mechanism leads to the more effective bandwidth utilization in comparison with protocols like CAN which delivers each data to the destination node separately [12, 16]. The better use of bandwidth results in faster update of data and therefore the average datalife time becomes lower.

The second reason is the slow data transfer rate and low performance of CAN in delivering messages with more than 8 bytes data size [6]. This leads to limited throughput for CAN and makes it unsuitable for transmission of messages with larger data sizes [6, 17, 18]. As we mention before, datalife time has an inverse relationship to the update rate of real-time data, so increase in update rate of real-time data result in datalife time reduction and consequently to the reduction of the expiration probability of data validity time.

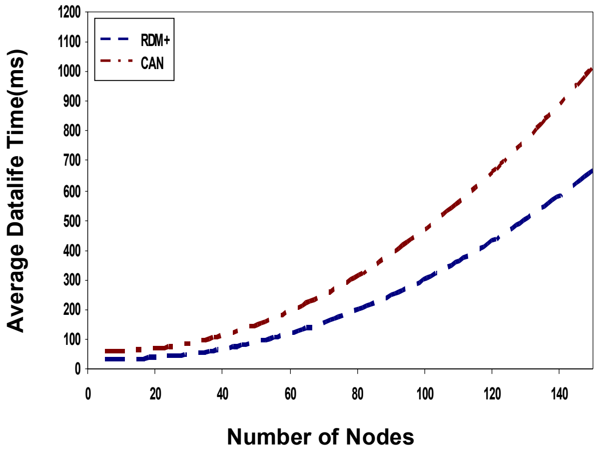

Figure 1. Average datalife time for messages with maximum 1000 bytes data size

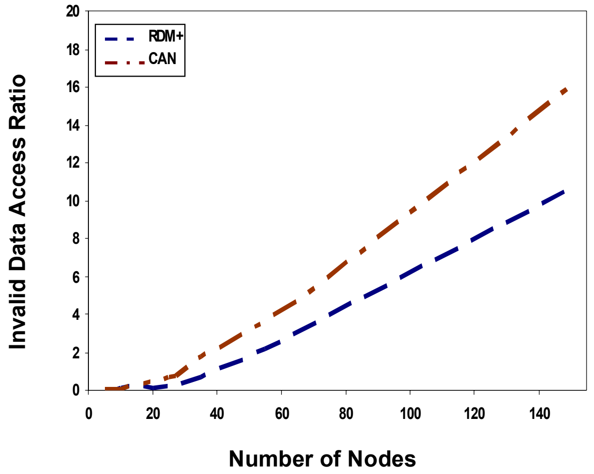

Figure 2. Invalid data access ratio for messages with maximum 1000 bytes data size

So in comparison with CAN, in RDM+ the chance of the accesses to be invalid will be decreased as it is shown in Figure 2.

In Figure 3 and 4, results of the average datalife time and invalid data access ratio for messages with size less than 8 bytes are shown. In this case RDM+ and CAN results get closer to each other in both metrics, but RDM+ still obtains better results. The main reason that causes these close results is the messages data size which is bounded to 8 bytes. CAN shows better performance and better throughput for passing messages with less than 8 bytes data size in comparison with messages with size larger than 8 bytes [6]. So, in this case CAN can update real-time data faster and therefore the life time of real-time data becomes shorter than previous case. But, still, the effective bandwidth utilization in RDM+ and the longer transmission time in CAN cause the higher update rate of the real-time data in RDM+ [6, 18, 19]. As it is obvious in Figure 3 and 4, both RDM+ and CAN obtain better results than previous case. The reason is that shorter messages can be delivered faster.

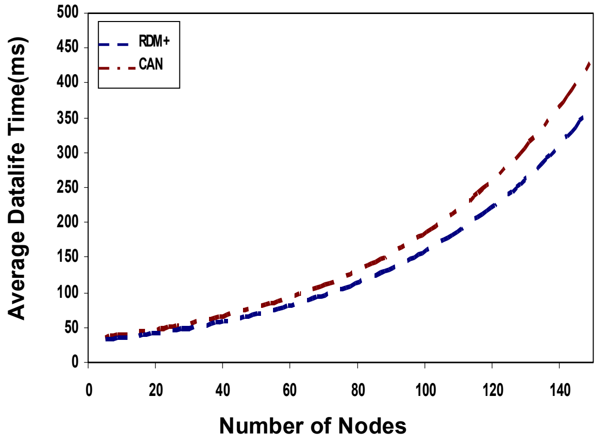

Figure 3. Average datalife time for messages with maximum 8 bytes data size

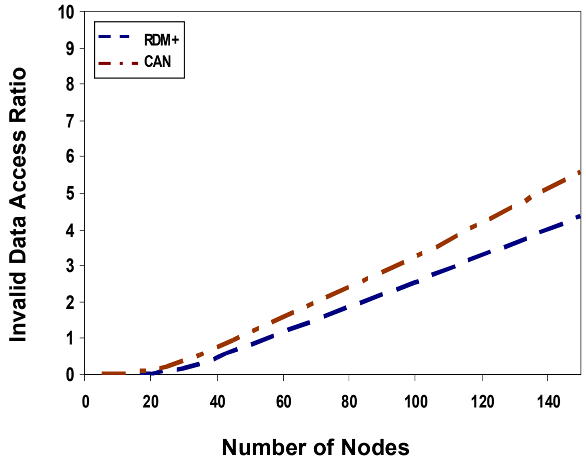

Figure 4. Invalid data access ratio for messages with maximum 8 bytes data size

As a general conclusion, RDM+ with its higher update rate and the lower invalid data access ratio can better match with real-time systems that insist on a short time intervals between two consecutive updates. Such systems require real-time data which are updated recently and do not desire to use out dated data.

5. CONCLUSION

In this paper we briefly described RDM message-passing mechanism and its impact on RDM+ protocol performance. We compared RDM+ with CAN to see which one has a better update rate. Invalid data access ratio and average data life time have been computed for both protocols and it has been shown that RDM+

produces fresher data and has the potential to be used in future distributed real-time systems.

6. REFERENCES

[1] M. Schofield, Controller Area Network, *How CAN Works*, 1996.

[2] M. Naghibzadeh, Round Data Mailer Message, *IEEE International Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia, Oct. 2002, pp. 586-591.

[3] M. Sabeghi, M. Naghibzadeh, K.L.M. Bertels, RDM+: A New Mac Layer Real-Time Communication Protocol, *IEEE Sarnoff Symposium*, April 2007, in press.

[4] S. Agrawal, P. Bhatt, *Real-time Embedded Software Systems -An Introduction*, Tata Consultancy Services, August 2001.

[5] M. Joseph, *Real-time Systems: Specification, Verification and Analysis*, Prentice Hall International, London, 1996.

[6] L. Feng-Li, J. R Moyne, D.M Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *Control Systems Magazine, IEEE* Volume 21, Issue 1, Feb 2001, pp. 66-83.

[7] C. Guo, Z. Guo, Q. Zhang, and W. Zhu, A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks, *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, June 2004, pp. 834-848.

[8] H. Kopetz, A. Damm, C. Koza. M. Mulazzani, W. Schwabl, C. Senft, and R. Zainlinger, "Distributed Fault-Tolerant Real-Time System," *The Mars Approach, IEEE Macro*, Feb. 1989, pp25-40.

[9] L. M. Pinho, F. Vasques and L. Ferreira, Programming Atomic Multicasts in CAN, *10th International Real-Time Ada Workshop (IRTAW'2000)*, Avila, Spain, Sept. 2000, pp18-22.

[10] ANSI X3T9.5, "*FDDI Token Ring Media Access Control*", (May 1987).

[11] F. Hanssen, P. G. Jansen, *Real-time communication protocols: an overview*, Technical report TR-CTIT-03-49, Centre for Telematics and Information Technology, University of Twente, Enschede, the Netherlands, October 2003.

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

[13] P. H. Welch, M. D. May, P. W. Thompson, *Networks, Routers and Transputers: Function, Performance and Application*, IOS Press, Netherlands, Feb. 1993, ISBN 90-5199-129-0.

[14] N. Malcolm, S. Kamat, W. Zhao, *Real-time communication in FDDI networks*, Real-Time Systems, Jan. 1996, pp. 75-107.

[15] M. Sabeghi, M. Naghibzadeh, Performance Assessment of a Distributed Real-Time Control System Utilizing RDM and RDM+ Protocols for Communication, *2nd International Conference on Future Networking Technologies (CoNEXT06),* ACM SIGCOMM, December 2006.

[16] S. Mirshokraie, M. Sabeghi, M. Naghibzadeh, K.L.M. Bertels, Performance Evaluation of Real-Time Message Delivery in RDM Algorithm, *IEEE Third International Conference on Networking and Services*, Athens, Greece, June 2007, in press.

[17] S. Biegacki, D. VanGompel, The application of DeviceNet in process control, *ISA Transactions*, vol. 35, no. 2, 1996, pp. 169-176.

[18] K. Tindell, A. Burns, A.J. Wellings, Calculating Controller Area Network (CAN) message response times, *Control Engineering Practice*, vol. 3, no. 8, Aug. 1995, pp. 1163-1169.

[19] K. W. Tindell, H. Hansson, A. J. Wellings, Analysing real-time communications: Controller Area Network (CAN), *15th IEEE Real-Time Systems Symposium*, San Juan, Puerto Rico, Dec. 1994, pp. 259-263, IEEE Computer Society Press, ISBN 0-8186-6600-5.