A Study on the Performance Improvement of Message Transmission over MVB(Multifunction Vehicle Bus)

Myung-Ho Choi*, and Jaehyun Park**

* Department of Information and Communication Engineering, Inha University, Incheon, Korea
(Tel : +82-32-863-0442; E-mail: mhchoi@resl.inha.ac.kr)
**Department of Information and Communication Engineering, Inha University, Incheon, Korea
(Tel : +82-32-860-7713; E-mail: jhyun@inha.ac.kr)

Abstract: The data transmission of MVB(Multifunction Vehicle Bus) of TCN(Train Network Communication) is divided into the periodic transmission phase and the sporadic transmission phase. TCN standard defines the event-polling method for the message transfer in the sporadic phase. However, since the event-polling method does not use pre-scheduling to the priority of the messages to be transmitted, it is inefficient for the real-time systems. To schedule message transmission, a master node should know the priority of message to be transmitted by a slave node prior to the scheduling the sporadic phase, but the existing TCN standard does not support any protocol for this. This paper proposes the slave frame bit-stuffing algorithm, with which a master node gets the necessary information for transmission scheduling and includes the simulation results of the event-polling method and the proposed algorithm.

Keywords: Train Communication Network, Message Transmission, Event-polling. Slave Frame Bit-stuffing

1. INTRODUCTION

With the development of the control technique of the electric railway train, the network system is recently used for the execution of various functions in a distributed environment. As a result, the importance of the network system grows gradually in control systems of the electric railway train. As a main control network, it should provide several key features; (1) both of intelligent stations and I/O devices should share a data bus (2) data that has time-critical characteristic like control signal and an amount of data that has weak time-critical characteristic must be able to be transmitted (3) self-configurability for the change of vehicle configuration is needed.

IEC(International Electro-technical Commission) proposed TCN(Train Communication Network), IEC 61375, coincided with above characteristics as the communication protocol of the electric railway. The main purposes of TCN are accurate data exchange in time and co-operation with flexibility among vehicles or plug-in devices [1,2,3].

In Fig. 1, TCN has a hierarchical structure that consists of MVB(Multifunction Vehicle Bus) and WTB(Wired Train Bus) as the intra-vehicle bus and the inter-vehicle bus, respectively [1,2,3,4,5,7].

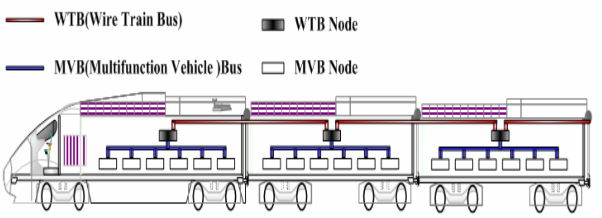

Fig. 1 The hierarchical structure of TCN

The data service of MVB has three kinds of services; process data service, message data service, and supervisory data service. As in Fig. 2, the time period of data transmission is divided into the periodic transmission phase and the sporadic transmission phase to the transmission method.

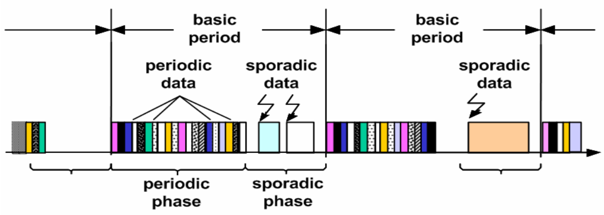

Fig. 2 Periodic data and sporadic data transmission

The message data service among the three data services of MVB uses the *Event-polling method* in the sporadic phase. However, in case of the event-polling method, it is impossible to schedule message data transmission in prior because it is impossible for a master node to know transmission priority. This paper brings up the problems of the event-polling-based message data transmission and introduces the *Slave Frame Bit-stuffing algorithm.*

In section 2, the basic data transmission method of MVB is described. In section 3, slave frame bit-stuffing algorithm is proposed. In section 4, the simulation results of the proposed algorithm and comparison to the event-polling method are shown. Section 5 concludes this paper.

2. DATA TRANSMISSION OF MVB(MULTIFUNCTION VEHICLE BUS)

2.1 Overview of MVB Frames

There are two kinds of MVB telegrams, the master frame generated by a master node and the slave frame generated by a slave node as a response to the master frame.

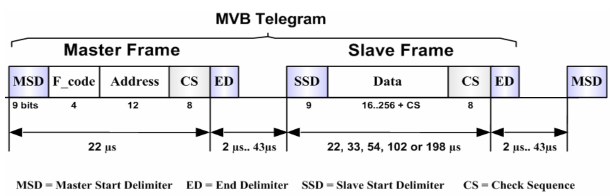

Fig. 3 MVB Telegram

All nodes decode a master frame when a master node broadcasts a master frame. After that, a slave node that is assigned as a source node of the decoded master frame transmits a slave frame. At this moment, the data service is determined by F_code of the master frame. The data service corresponded to each F_code is presented at the Table 1.

Table 1 F_codes of the MVB(Master Frame Types).

F_code	Telegram Type
0	Process Data Request with size 16 bits
1	Process Data Request with size 32 bits
2	Process Data Request with size 64 bits
3	Process Data Request with size 128 bits
4	Process Data Request with size 256 bits
8	Mastership Transfer Request
9	General Event Request
12	Message Data Request with 256 bits
13	Group Event Request
14	Single Event Request
15	Device Status Request
5, 6, 7, 10, 11	(reserved)

2.2 Message Data Transmission over MVB

TCN standard recommends that the bandwidth of periodic transmission phase of MVB should not exceed 60% of a basic period. In the periodic transmission phase, the process data service and supervisory data service are served. A master node broadcasts master frames that are scheduled in advance. At this time, the address used in master frame is not the physical node address but the logical port address. Each logical port address has the individual period that is determined with considering several factors; the priority, the data refresh period, and importance of data.

In the sporadic transmission phase, the message data service and supervisory data service are served. Unlike the process data service, the master frame of message data service uses not the logical port address but the physical node address. The F_code is also fixed 12. The message transmission process follows *caller/replier* model that is similar to *client/server* model. The messages are able to be divided into several packets, and each packet has the destination address. Each node has the physical node address to receive a message and has the transmission and receipt queues. When a message data occurs, the pertinent node transmits a request of message transmission to a master node. This on-demand request policy draws a critical disadvantage because a master node isn't able to know which node has the message data to transmit in advance. To find out a slave node to transmit a message, TCN uses collision-based event-polling method.

2.3 Event-polling method

The Event-polling method uses the *binary-search-tree* algorithm, which is used as MAC(Media Access Control) protocol to resolve collisions that may happen in the sporadic transmission phase of MVB. In case of symmetric protocol, each node should examine all feedback during CRP(Collision Resolution Period). But in case of asymmetric protocol, such as MVB, only a master node examines feedback.

A master node using a binary-search algorithm has one of the four states; *Response*, *Silence*, *Collision* and *One-answer*[6].

- **General Event Request**: As the start of Event-polling, the state to take the message transmission requests.
- **Group Event Request**: The state to poll a group of nodes belonging to a group.
- **Single Event Request**: The state to poll a single node for its event when response by General Event Request or Group Event Request is One answer.
- **Message Transfer Request**: As polling the physical node address of response about Single Event Request, the state to demand the message transmission.

Fig 4 shows the example of event-polling method with 8 nodes that have the physical node address of 3 bits. It is remarkable to lower on level for fast search when *Silence* occurs.

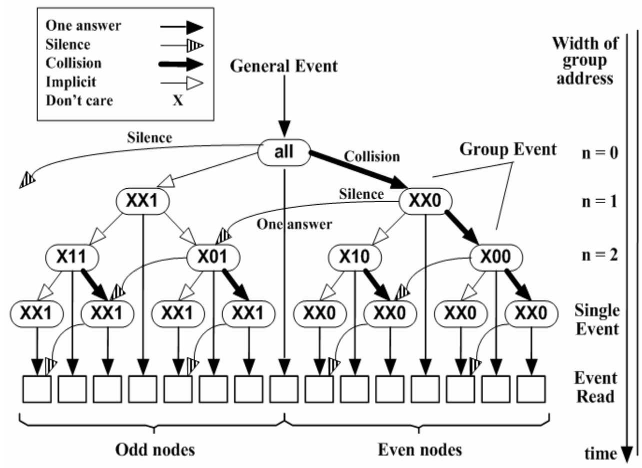

Fig. 4 Event Arbitration Tree

When a collision occurs, the time delay of event-polling is much longer due to the tree search as the number of nodes increases. Besides, in case of priority of a message is considered, if response for the high-priority request is *Silence*, a master node must poll nodes for low-priority request again. This situation may cause unnecessary waste of network bandwidth and time delay. Moreover, if the physical node address of each node is not assigned continuously, the worst case of delay is predicted when the address of two nodes differs only in their MSB(Most significant Bit). In this the worst case, 14 telegrams(1 General Event, 11 Group Event, 1 Single Event, 1 Event Read) are required to read the first event. This long time delay may not acceptable in time-critical real-time systems. Table 2 indicates the arbitration delay.[1]

Table 2 Arbitration delay in function of the number of simultaneous events

Simultaneous Events	Average frame number	Maximum frame number (12-bit address)
0	1	1

1	2	2
2	5.0	15
3	8.3	16
4	12	29
8	16	30

To reduce the arbitration time delay, TCN standard recommends assigning consecutive addresses, beginning with 1. But this solution is not flexible because it is not considered of plug-in devices and devices co-operation.

3. SLAVE FRAME BIT-STUFFING

If a master node has the information of message transmission before the sporadic transmission phase, a master node can consider transmission priority and avoid time delay caused by the event-polling method. For this, nodes need to transmit their information of message transmission to a master node before the sporadic transmission phase.

This paper proposes *Slave-Frame-Bit-stuffing* method that stuffs slave frame with necessary information for message data transmission by bit unit. TCN standard defines a time interval between transmitted slave frame and next master frame to minimum 3μs (if it expects neither collision nor silence in response to its previous master frame). After transmission of slave frame ends, when state of MVB is idle, the time to be required is maximum 300ns in three kind of physical layers(Electric Short Distance, Electric Middle Distance, Optical Glass Fibre). That is, Slave Frame Bit-stuffing must begin between minimum 300ns and maximum 3μs.

The added information by Slave Frame Bit-stuffing method includes the existence of message data from the designated slave node and its priority. To inform the existence of message data, at leaset 12 bits are required to identify the physical node address. In addition to this, one bit of priority bit is required. Besides, one starting bit, always 1, is necessary. As a result, total 14 bits are needed for Slave Frame Bit-stuffing method. The frame used by Slave Frame Bit-stuffing method has the format shown in Fig. 5. The meaning of each bit field and time variables are as follows.

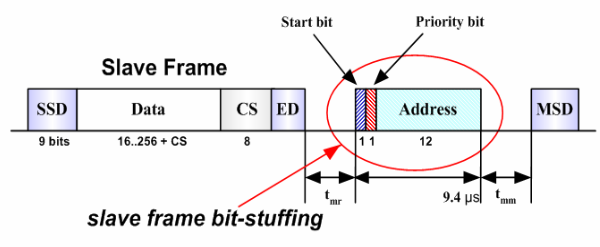

Fig. 5 The frame of Slave Frame Bit-stuffing

- bit 1: If message data to transmit exits, set ‘1’
- bit 2: If message data to transmit is high priority, set ‘1’. If message data to transmit is low priority, set ‘0’
- bit 3~14: If message data to transmit exits, set the self physical node address
- t_{mr}: The time interval between the transmitted slave frame and next frame of Slave Frame Bit-stuffing. We recommend 2μs
- t_{mm}: The time interval between transmitted frame of Slave Frame Bit-stuffing and next master frame. We recommend 2μs

A master node can schedule message data transmission without arbitration after decoding frames applied Slave Frame Bit-stuffing method during the periodic transmission phase. The following Table 3 shows the delay generated by frames applied Slave Frame Bit-stuffing method.

Table 3 Delay in function of the number of simultaneous events

Simultaneous Events	frame number (12-bit address)
0	0
1	1
2	2
3	3
4	4
8	8

When compared Table 2 with Table 3, event-polling requires much more frames wasted for arbitration as the number of simultaneous events increases. This means that time delay is large to process events. On the other hand, in the case of Slave Frame Bit-stuffing method, the time delay is smaller than that of event-polling method because the number of used frames equals number of simultaneous events.

Besides, in case of event-polling, unnecessary frames are generated because it uses event request twice to pick out the priority of message data transmission. Compared to event-polling, the frames applied Slave Frame Bit-stuffing have the information of the priority of message data transmission without any extra polling. This means to reduce the time delay.

It is possible to schedule message data transmission in prior by using the information of frames from Slave Frame Bit-stuffing method. It is big advantage to operation and flexibility from the viewpoint of network management.

4. SIMULATION

To examine the performance of the proposed algorithm, event-polling and Slave Frame Bit-stuffing method are modeled and simulated by ARENA, a well-known commercial simulation software[13]. The assumptions and parameters used for the simulation are summarized as follows.

- The number of nodes is restricted to 8.
- In the case of Slave Frame Bit-Stuffing method, a master node uses binary search tree algorithm which is used in Event-polling method as message data scheduling algorithm.
- A mastership transfer can’t occur.
- Process data service can’t exceed 60% of basic period. For this, number of process data transmissions in each basic period is limited to 10 and F-code of all process data is set to 1(16bit).
- Data transmission time of MVB communication is defined as follows based on TCN standard. Then, we applied these conditions to both scenarios.
  - Master frame transmission time: 22μs
  - Slave frame of process transmission time: 22μs

- Message request time: 22μs
- Slave frame of request response time: 22μs
- Slave message frame transmission time: 198μs

- Consecutive physical addresses are assigned to minimize time delay of message transmission on Event-polling method.
- The troubles of communication can't occur

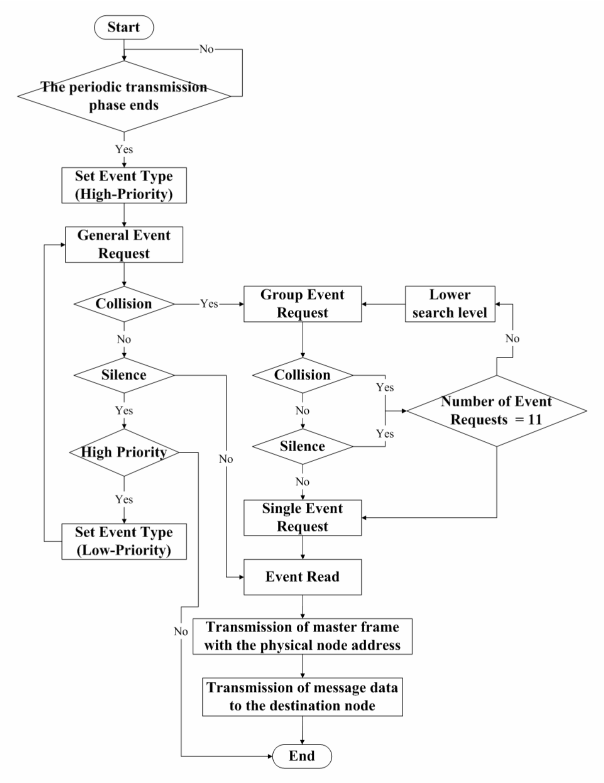

Fig. 6 Modeling of Event-polling method

Fig. 8 and Fig. 9 show the number of messages processed by event-polling and Slave Frame Bit-stuffing method respectively. In these figures, the number of messages processed by Slave Frame Bit-stuffing method is far more than that by Event-polling method.

Fig. 10 shows that comparisons of time delay between two methods, when number of messages processed of each method are same. In the same manner, time delay of Slave Frame Bit-stuffing method is less than time delay of event-polling method.

In the case of event-polling method, consecutive addresses are used in this simulation. However if consecutive addresses are not used, the time delay of binary search tree algorithm may increase. On the other hand, the performance Slave Frame Bit-stuffing method is not changed because no delay exists by address allocation.

From the simulation results, the proposed algorithm in this paper, the Slave Frame Bit-stuffing method, improves the performance of the message transmission over MVB.

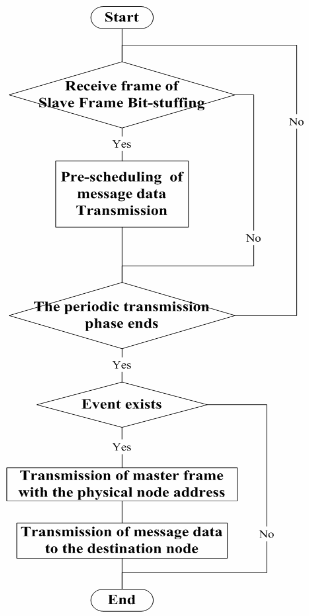

Fig. 7 Modeling of Slave Frame Bit-stuffing method

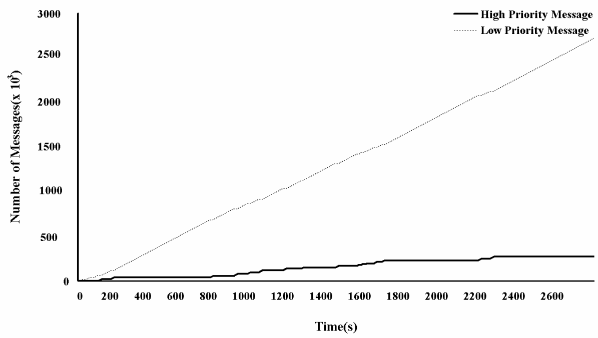

Fig. 8 Throughput of message transmission for the Even-polling based MVB

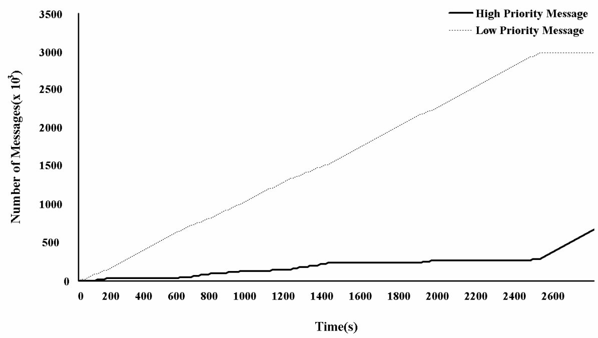

Fig. 9 Throughput of message transmission for the Slave Frame Bit-stuffing based MVB

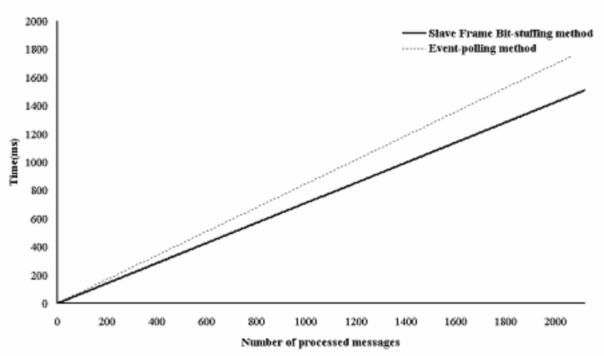

Fig. 10 A comparison of the time delay for the two message transmission methods

5. CONCLUSION

This paper proposes the *Slave Frame Bit-stuffing* method to improve the performance of message data transmission over TCN-MVB and presents performance improvement of message data transmission by computer simulation. The performance of message data transmission and bandwidth utilization are improved by using Slave Frame Bit-stuffing. In case of using event-polling method, if consecutive addresses are assigned(statistic addressing method), the performance of message data transmission improves a little. Some bandwidth loss is expected during process data services when Slave Frame Bit-stuffing method is used. To minimize this bandwidth loss, a further research on the dynamic scheduling of process data is expected[8].

ACKNOWLEDGMENTS

This research is partially supported by the G7 project, Korean High-speed Train Development Project sponsored by Ministry of Construction and Transportation, Ministry of Science and Technology, and Ministry of Commerce, Industry and Energy.

REFERENCES

[1] IEC 61375-1 Standard, Train Communication Network: Part *(1)General Architecture (2)Real-time Protocol (3)Multifunction Vehicle Bus (4)Wire Train Bus (5)Train Network Management (6)Train Communication Conformance Testing*, 1999.
[2] UIC 556 Standard, Information Transportation on the Train Bus, 1999. 05. 01. v2.0
[3] H. Kirrmann and P. A. Zuber, “IEC/IEEE Train Communication Network,” 1996.
[4] G. Fadin and F. Cabaliere, “IEC TC9 WG22 Train Communication Network Dependability and Safety concepts,” *World Congress on Railway Research 97*, 1997.
[5] P. Etter and H. Kirrmann, “Use of standardized integrated communication systems on vehicles and trains,” *Proceedings of the International Conference on Speedup Technology for Railway and Maglev Vehicles*, Vol.1, 1993.
[6] J. H. Park and S. C. Lee, “Performance Evaluation of the Train Communication Network,” *Workshop on Distributed Computer Control Systems*, 1998
[7] C. Schifers and G. Hans, “IEC 61375-1 and UIC 556-international standards for train communication,” *Vehicular Technology Conference Proceedings*, VTC 2000-Spring Tokyo. 2000 IEEE 51st , Volume: 2 , 2000
[8] A. Chavarria, J.L. de Arroyabe, A. Zuloaga, J. Jimenez, J.L. Martin, G. Aranguren, “Slave node architecture for train communications networks,” *IEEE Conference*, Vol.4 , 2000
[9] S. Cavalieri, A. Di Stefano and O. Mirabella, “Acyclic Traffic Management in FieldBus Scenarios,” *IEEE International Workshop*, 1992.
[10] S. Cavalieri and O. Mirabella, “Enhancing Performance in FieldBus Communication Systems,” *IEEE Conference*, Vol.2, 1996
[11] Y. S. Kim and W. H. Kwon, “Pre-runtime Scheduling Methods of Data link layer in the Fieldbus environment,” *Workshop on Distributed Computer Control Systems*, 1997
[12] L.R.H.R Franco, “Transmission scheduling for fieldbus: a strategy to schedule data and messages on the bus with end-to-end constraints,” *IEEE International Joint Symposia*, 1996
[13] W. David Kelton, Randall P. Sadowski and Deborah A. Sadowski, “Simulation with ARENA,” McGraw-Hill, Inc., 1998.